UNIVERSIDAD DE PIURA

Book of Syllabi

School of Computer Science

– 2022-I–

Lima: January 24, 2021

Task Force

Ernesto Cuadros-Vargas (Editor) <*ecuadros@spc.org.pe*>
President of the Peruvian Computer Society (SPC) 2001-2007, 2009
Member of the Steering Committee de ACM/IEEE-CS Computing Curricula for Computer Science (CS2013)
Member of Steering Committee de ACM/IEEE-CS Computing Curricula 2020 (CS2020)
http://socios.spc.org.pe/ecuadros

Contents

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS111. Computing Foundations (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the first course in the sequence of introductory courses to Computer Science.This course is intended to cover the concepts outlined by the Computing Curricula IEEE-CS/ACM 2013. Programming is one of the pillars of Computer Science; any professional of the area, will need to program to materialize their models and proposals. This course introduces participants to the fundamental concepts of this art. Topics include data types, control structures, functions, lists, recursion, and the mechanics of execution, testing, and debugging.

5. GOALS

- Introduce the fundamental concepts of programming.
- Develop the ability of abstraction using programming language

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: History (5)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Prehistory, the world before 1946 • History of computer hardware, software, networking • Pioneers of computing • History of the Internet	• Identify significant continuing trends in the history of the computing field [Familiarity] • Identify the contributions of several pioneers in the computing field [Familiarity] • Discuss the historical context for several programming language paradigms [Familiarity] • Compare daily life before and after the advent of personal computers and the Internet [Assessment]
Readings : [BB19], [Gut13], [Zel10]	

<table>
<tr><td colspan="2">Unit 2: Basic Type Systems (2)</td></tr>
<tr><td colspan="2">Competences Expected: a</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

A type as a set of values together with a set of operations

Primitive types (e.g., numbers, Booleans)
Compound types built from other types (e.g., records, unions, arrays, lists, functions, references)

Association of types to variables, arguments, results, and fields
Type safety and errors caused by using values inconsistently given their intended types

</td><td>

For both a primitive and a compound type, informally describe the values that have that type [Familiarity]
For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity]
Describe examples of program errors detected by a type system [Familiarity]
For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage]
Use types and type-error messages to write and debug programs [Usage]
Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Gut13], [Zel10]</td></tr>
</table>

Unit 3: Fundamental Programming Concepts (9)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment] • Identify and describe uses of primitive data types [Familiarity] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Familiarity] • Describe the concept of recursion and give examples of its use [Assessment] • Identify the base case and the general case of a recursively-defined problem [Familiarity]
Readings : [Gut13], [Zel10]	

Unit 4: Basic Analysis (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Big O notation: use • Analysis of iterative and recursive algorithms	• Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm [Familiarity] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Familiarity] • State the formal definition of big O [Familiarity] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage]
Readings : [Gut13], [Zel10]	

<table>
<tr><td colspan="2">Unit 5: Fundamental Data Structures and Algorithms (8)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max,
• Sequential and binary search algorithms
• Worst case quadratic sorting algorithms (selection, insertion)
• Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)
• Hash tables, including strategies for avoiding and resolving collisions
• Binary search trees
  – Common operations on binary search trees such as select min, max, insert, delete, iterate over tree
• Graphs and graph algorithms
  – Representations of graphs (e.g., adjacency list, adjacency matrix)
  – Depth- and breadth-first traversals
• Heaps
• Graphs and graph algorithms
  – Maximum and minimum cut problem
  – Local search
• Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)
</td><td>
• Implement basic numerical algorithms [Usage]
• Implement simple search algorithms and explain the differences in their time complexities [Assessment]
• Be able to implement common quadratic and O(N log N) sorting algorithms [Usage]
• Describe the implementation of hash tables, including collision avoidance and resolution [Familiarity]
• Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity]
• Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity]
• Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity]
• Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage]
• Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment]
• Describe the heap property and the use of heaps as an implementation of priority queues [Familiarity]
• Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage]
• Trace and/or implement a string-matching algorithm [Usage]
</td></tr>
<tr><td colspan="2">Readings : [Gut13], [Zel10]</td></tr>
</table>

Unit 6: Algorithms and Design (9)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• The concept and properties of algorithms
  – Informal comparison of algorithm efficiency (e.g., operation counts)
• The role of algorithms in the problem-solving process
• Problem-solving strategies
  – Iterative and recursive mathematical functions
  – Iterative and recursive traversal of data structures
  – Divide-and-conquer strategies
• Fundamental design concepts and principles
  – Abstraction
  – Program decomposition
  – Encapsulation and information hiding
  – Separation of behaivor and implementation
	• Discuss the importance of algorithms in the problem-solving process [Familiarity]
	• Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity]
	• Create algorithms for solving simple problems [Usage]
	• Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage]
	• Implement, test, and debug simple recursive functions and procedures [Usage]
	• Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment]
	• Implement a divide-and-conquer algorithm for solving a problem [Usage]
	• Apply the techniques of decomposition to break a program into smaller pieces [Usage]
	• Identify the data components and behaviors of multiple abstract data types [Usage]
	• Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage]
	• Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]

Readings : [Gut13], [Zel10]	

Unit 7: Development Methods (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Modern programming enviroments
  – Code search
  – Programming using library components and their APIs
	• Construct and debug programs using the standard libraries available with a chosen programming language [Familiarity]

Readings : [Gut13], [Zel10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview*. Ed. by PEARSON. Global Edition. Pearson, 2019. ISBN: 1292263423. URL: `http://www.pearsonhighered.com/brookshear`.

[Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python*. MIT Press, 2013.

[Zel10] John Zelle. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2010.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS1D1. Discrete Structures I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Discrete structures provide the theoretical foundations necessary for computation. These fundamentals are not only useful to develop computation from a theoretical point of view as it happens in the course of computational theory, but also is useful for the practice of computing; In particular in applications such as verification, cryptography, formal methods, etc.

5. GOALS

- Apply Properly concepts of finite mathematics (sets, relations, functions) to represent data of real problems.
- Model real situations described in natural language, using propositional logic and predicate logic.
- Determine the abstract properties of binary relations.
- Choose the most appropriate demonstration method to determine the veracity of a proposal and construct correct mathematical arguments.
- Interpret mathematical solutions to a problem and determine their reliability, advantages and disadvantages.
- Express the operation of a simple electronic circuit using Boolean algebra.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Sets, Relations, and Functions (22)

Competences Expected: a,j

Topics

- Sets
  - Venn diagrams
  - Union, intersection, complement
  - Cartesian product
  - Power sets
  - Cardinality of finite sets
- Relations:
  - Reflexivity, simmetry, transitivity
  - Equivalence relations
  - Partial order relations and sets
  - Extremal elements of a partially ordered sets
- Functions
  - Surjections, injections, bijections
  - Inverses
  - Composition

Learning Outcomes

- Explain with examples the basic terminology of functions, relations, and sets [Assessment]
- Perform the operations associated with sets, functions, and relations [Assessment]
- Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context [Assessment]

Readings : [Gri03], [Ros07], [Vel06]

Unit 2: Basic Logic (14)

Competences Expected: a,j

Topics

- Propositional logic
- Logical connectives
- Truth tables
- Normal forms (conjunctive and disjunctive)
- Validity of well-formed formula
- Propositional inference rules (concepts of modus ponens and modus tollens)
- Predicate logic
  - Universal and existential quantification
- Limitations of propositional and predicate logic (e.g., expressiveness issues)

Learning Outcomes

- Convert logical statements from informal language to propositional and predicate logic expressions [Usage]
- Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms [Usage]
- Use the rules of inference to construct proofs in propositional and predicate logic [Usage]
- Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (eg, program correctness), database queries, and algorithms [Familiarity]
- Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles [Usage]
- Describe the strengths and limitations of propositional and predicate logic [Usage]

Readings : [Ros07], [Gri03], [Vel06]

Unit 3: Proof Techniques (14)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction • The structure of mathematical proofs • Direct proofs • Disproving by counterexample • Proof by contradiction • Induction over natural numbers • Structural induction • Weak and strong induction (i.e., First and Second Principle of Induction) • Recursive mathematical definitions • Well orderings	• Identify the proof technique used in a given proof [Assessment] • Outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit [Usage] • Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument [Usage] • Determine which type of proof is best for a given problem [Assessment] • Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures [Familiarity] • Explain the relationship between weak and strong induction and give examples of the appropriate use of each [Assessment] • State the well-ordering principle and its relationship to mathematical induction [Familiarity]
Readings : [Ros07], [Sch12], [Vel06]	

Unit 4: Data Representation (10)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Numerical representation: sign-magnitude, floating point. • Representation of other objects: sets, relations, functions.	• Explain numerical representations such as sign-magnitude and floating point. [Assessment]. • Carry out arithmetic operations using different kinds of representations. [Assessment]. • Explain the floating point standard IEEE-754 [Familiarity].
Readings : [Ros07], [Gri03], [Vel06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

[Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction.* 3 ed. Brooks Cole, 2012.

[Vel06] Daniel J. Velleman. *How to Prove It: A Structured Approach.* Ed. by Cambridge University Pres. 2nd. 2006. ISBN: 978-0521675994.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

MA100. Mathematics I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	5
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The course aims to develop in students the skills to deal with models in science and engineering related to single variable differential calculus skills. In the course it is studied and applied concepts related to calculation limits, derivatives and integrals of real and vector functions of single real variables to be used as base and support for the study of new contents and subjects. Also seeks to achieve reasoning capabilities and applicability to interact with real-world problems by providing a mathematical basis for further professional development activities.

5. GOALS

- Apply knowledge of mathematics.
- Apply engineering knowledge.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Vectors and complex numbers (20)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Operations with complex numbers • Theorem Moivre	• Define and operate with complex numbers, calculating their polar and exponential shape. • Use Moivre theorem to simplify complex calculations. • Operate with vectors by characterizing them by their direction and magnitude. Represent a function from the relation of sets, given verbally, graphically and algebraically, in a Venn diagram and/or in the Cartesian plane providing, if possible, its correspondence rule and its main characteristics.
Readings : [Ste12], [Lar18]	

Unit 2: Functions of a variable (10)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Definition, characteristics and graphic representation. • Function algebra. • Linear, polynomial, sinusoidal, exponential and logarithmic functions. • Modeling of situations close to reality using functions.	• Model real situations of the near environment using constant, linear, quadratic and polynomial functions, and others resulting from operations (f+-*/g, fog , af(bx -c)+d) between elementary functions, with emphasis on calculation, graphing and interpretation of slope and concavity in an applied context • Model real-life situations in the immediate environment using sine wave functions. • Use the exponential, logarithmic and logistic functions to model real situations of the near environment that adjust to their behavior, recognizing their characteristics (growth, decrease, asymptotic behavior). • Recognizes and builds trigonometric functions. • Aplicar reglas para transformar funciones.
Readings : [Ste12], [Lar18]	

Unit 3: Derivatives of functions (20)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Definition of derivative as rate of change and as slope of the tangent to the curve at a point. • Referral rules. • Applications of derivadees in related speed problems. • Applications of derivatives in function optimization problems.	• Solve problems using the derivative of a function as a ratio of change between its two variables or as the slope of the tangent line at a point, applying the derivation rules to simple functions. • Approximate functions using the differentials. $df = f'(x)dx$, applying the derivation rules to calculate derivatives of compound and implicit functions with Leibniz notation. • To solve real context problems of the near environment that involve the calculation of related speeds by deriving simple, compound functions and implicitly taking into account the use of differentials. • Solve optimization problems by analyzing the behavior of a function through its first and second derivatives (growth, decrease, concavity, extremes)
Readings : [Ste12], [Lar18]	

Unit 4: Integral (22)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Indefinite integral and integration methods (substitution, integration by parts, trigonometric substitutions and decomposition by partial fractions). • Riemann sum to estimate areas. • Calculation theorems (TFC1, TFC2, TCN). • Calculation of area between curves and average value. • Differential equations that are solved by separable variables.	• Solve undefined integrals by various methods (substitution, integration by parts, trigonometric substitution, decomposition into partial fractions). • Estimate the area under a curve by dividing it into Riemann rectangles and sums, with interpretations in physics and other everyday contexts. • Apply the calculation theorems (TFC1, TFC2, TCN) to solve undefined integrals using different integration methods. • Solve area and average value problems of a function, with the corresponding physical interpretations of the integral in kinematics. • Model real situations using differential equations and solve them using variable separation method (Newton's Cooling Law, Population Dynamics (Logistics, learning curve), etc.). • It defines a complex number and represents it in various ways. It uses Moivre's formula to calculate operations with complexes.
Readings : [Ste12], [Lar18]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Lar18] Ron Larson. *Cálculo*. Ed. by Cengage Learning. 10th. 2018.

[Ste12] James Stewart. *Cálculo de una variable Trascendentes tempranas*. Ed. by Cengage Learning. 7th. 2012. ISBN: 978-607-481-881-9.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG101. Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

To achieve an effective communication in the personal and professional field, The adequate handling of the language in oral and written form is a priority. It is therefore justified that the students know, understand and apply the conceptual and operational aspects of their language, for the development of their fundamental communication skills: listening, speaking, reading and writing.

Consequently, the permanent exercise and the contribution of the contribute greatly to academic training and, in the future in the course of their work

5. GOALS

- Develop communication skills through the theory and practice of language that help students to overcome the academic demands of the undergraduate program and contribute to their humanistic formation and as human beings.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (16)	
Competences Expected: C17,C20	
Topics	**Learning Outcomes**
• The communication, definition, relevance. Elements. Process. Functions. Classification. Oral and written communication. • The language: definition. Features and functions. Language: levels. System. Rule. Speaks. The linguistic sign: definition, characteristics. • Multilingualism in Peru. Dialect variations in Peru. • The word: definition, classes and structure. The monemas: lexema and morpheme. The morpheme: classes. Etymology. • The Academic Article: Definition, structure, choice of topic, delimitation of the topic.	• Recognize and value communication as a process of understanding and exchanging messages, differentiating its elements, functions and classification[Usage]. • Analyze the characteristics, functions and elements of language and language [Usage]. • Identify the characteristics of multilingualism in Peru, valuing its idiomatic richness [Usage]. • Identify the qualities of the word and its classes [Usage].
Readings : [Len10]	

Unit 2: (16)	
Competences Expected: C17, C24	
Topics	**Learning Outcomes**
• Paragraph: Main, secondary and global idea. • The text: definition, characteristics. Cohesion and coherence. • Organization of the text: The reference (dejis); Anaphora, cataphora, ellipsis. Logical and textual connectors. • Types of text: descriptive (processes), expository, argumentative. • Functions of elocution in the text: generalization, identification, nominalization, classification, exemplification, definition. • Discontinuous texts: graphs, tables and diagrams. • Search for information. Information sources. References and citations. Record of information: index cards, notes, summaries, etc. Critical apparatus: concept and purpose. APA Standards or other.	• Writing expository texts highlighting the main and secondary idea. [Usage]. • Write expository texts with adequate cohesion and coherence, making use of textual references and connectors. [Usage]. • Interpreting discontinuous texts, assessing their importance for the understanding of the message. [Usage].
Readings : [Len10], [Gat07]	

Unit 3: (12)	
Competences Expected: C17	
Topics	**Learning Outcomes**
• Prayer: definition and classes. The enunciative, interrogative, imperative, exclamatory and optional sentence. The proposition and the sentence. The simple and compound sentence. Coordination and subordination. The syntagm: structure and classes: nominal, verbal, adjectival, prepositional, adverbial. • Preparation of a glossary of technical terms, abbreviations and acronyms related to the specialty (permanent activity throughout the semester). • Writing the academic article: Summary, key words, introduction, development, conclusions, bibliographyTechnology (APA standards or other required by the Professional School).	• Recognizing and analyzing sentence structure, assessing its importance and usefulness in writing texts.[Usage]. • Register and use specialty-specific terminology. [Usage].
Readings : [San05]	

Unit 4: (12)	
Competences Expected: C17, C20, C24	
Topics	**Learning Outcomes**
• Writing correspondence: letter - application, report, memorandum, resume. • Oral speech: purposes, parts. Listening: purposes and conditions. Vices of diction: barbarism, solecism, cacophony, redundancy, amphibology, monotony. Prepositional regime. • Group communication Process, dynamics, structure Forms (Techniques): Round table, panel, forum and debate • Final review of the academic article. Presentation and oral presentation of intellectual production works.	• To write academic and functional texts taking into account the different moments of their production, their structure, purpose and formality. [Usage]. • Demonstrate skills as a sender or receiver in different communication situations with language correction. [Usage]. • Apply the different forms (techniques) of group communication recognizing their importance for problem solving, decision making or discussion. [Usage].
Readings : [Mar06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Gat07] Carlos Gatti Muriel. *Elementos de la gramática española*. Lima, Universidad del Pacífico., 2007.

[Len10] Real Academia de la Lengua Española. *Nueva gramática de la lengua española, morfología y sintaxis*. Madrid, España: Ed. Espasa, 2010.

[Mar06] Gonzalo Martin Vivaldi. *Teoría y práctica de la composición y estilo*. Thompson, 2006.

[San05] J Sanchez Lobato. *Saber Escribir*. España, Instituto Cervantes, 2005.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG102. Study Methodology (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Students in vocational training need to improve their attitude towards academic work and demands. In addition, they should understand the mental process that occurs in the exercise of study to achieve learning, so they know where and how to make the most appropriate adjustments to their needs. They also need to master various forms of study, so that they can select the strategies best suited to their personal learning style and the nature of each subject. They also need to know and use ways to search for academic information and do creative work of a formal academic nature, so that they can apply them to their college work, making their effort successful.

5. GOALS

- Develop in the student attitudes and skills that promote autonomy in learning, good academic performance and their training as a person and professional.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (12)	
Competences Expected: C19, C24	
Topics	**Learning Outcomes**
• The underlining. • Stitch taking. • Vocation, habits of university life. • Human interaction. • The will as a requirement for learning. • Planning and time.	• To analyze the normative documentation of the University evaluating its importance for the coexistence and academic performance. [Usage] • Understand and value the demands of university life as part of personal and professional training.[Usage] • Properly plan your time based on your personal and academic goals.[Usage] • Develop a personal improvement plan based on self-knowledge.[Usage]
Readings : [bibliografiaTecnologia]	

Unit 2: (12)	
Competences Expected: C19,C24	
Topics	**Learning Outcomes**
• Summary. Notes in the margin. Mnemonics. • Mental processes: Simple, complex. Fundamentals of meaningful learning. • The steps or factors for learning. Laws of learning. Learning style questionnaire Identification of personal learning style. • Academic reading. Levels of analysis of a text: central idea, main idea and secondary ideas. Meza de Vernet's model. • Exams: Preparation. Guidelines and strategies before, during and after an exam. Emotional intelligence and exams. • The sources of information. Critical device: concept and purpose. Vancouver standards. References and quotations.	• Identify mental processes by relating them to learning. [Usage]. • Understand the learning process to determine your own style and incorporate it into your academic activity. [Usage]. • Develop strategies for text analysis by enhancing reading comprehension. [Usage]. • To design a strategic program to successfully face the exams.[Usage].
Readings : [Rod07], [Per10], [Qui07]	

Unit 3: (12)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• The concept maps. Characteristics and elements. • Copyrights and plagiarism. Personal or moral rights. Economic rights. "Copyrigth". • Self-esteem, Emotional Intelligence, Assertiveness and Resilience. Concepts, development and strengthening. • Critical Apparatus: Vancouver Standards. Practical application. • Generation of ideas. Strategies for organizing ideas, writing and reviewing.	• To apply the techniques of study taking into account their particularities and adapting them to the different situations demanded by the learning. [Usage]. • Recognize the importance of respect for intellectual property. [Usage]. • Recognize the importance of EQ, assertive behavior, self-esteem and resilience by valuing them as strengths for college performance. [Usage].
Readings : [Chá11], [Vel99]	

Unit 4: (12)	
Competences Expected: C19	
Topics	**Learning Outcomes**
• Synoptic Table. The mind maps. Practice with the subject matter of the course. • The personal method of study. • The cooperative learning: definition, study groups, organization, members' roles. • Guidelines to conform efficient and harmonic groups. • The personal study method. Reinforcement of study techniques. • Presentation and exposition of works of intellectual production. • The debate and the argumentation.	• To apply the techniques of study taking into account their particularities and adapting them to the different situations demanded by the learning. [Usage]. • Assume management of behaviors and attitudes for cooperative learning and performance in work teams. [Usage]. • Formulate a personal study method project, according to your style and needs, including techniques and strategies. [Usage].
Readings : [Rod07], [Chá11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Chá11] A. Chávez. *Se necesita un tutor*. UCSP, 2011.

[Per10] A.E. Perez. *Teoría del Derecho*. Editorial Madrid, 2010.

[Qui07] V. Quintana. *El estudio Universitario y elementos de investigación científica*. Editorial universitaria, 2007.

[Rod07] J. Rodríguez. *Guía para el método de estudio universitario*. Educa, 2007.

[Vel99] Marco Flores Velazco. *Mapas conceptuales en el aula*. Ed. San Marcos, 1999.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS112. Computer Science I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	5
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS111. Computing Foundations. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the second course in the sequence of introductory courses in computer science. The course will introduce students in the various topics of the area of computing such as: Algorithms, Data Structures, Software Engineering, etc.

5. GOALS

- Introduce the student to the foundations of the object orientation paradigm, allowing the assimilation of concepts necessary to develop information systems.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: General overwiew of Programming Languages (1)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Brief review of programming paradigms. • Comparison between functional programming and imperative programming. • History of programming languages.	• Discuss the historical context for several programming language paradigms [Familiarity]
Readings : [Str13], [Dei17]	

Unit 2: Virtual Machines (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• The virtual machine concept. • Types of virtualization (including Hardware/Software, OS, Server, Service, Network). • Intermediate languages.	• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment]
Readings : [Str13], [Dei17]	

<table>
<tr><td colspan="2">Unit 3: Basic Type Systems (2)</td></tr>
<tr><td colspan="2">Competences Expected: a,b,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

A type as a set of values together with a set of operations

Primitive types (e.g., numbers, Booleans)
Compound types built from other types (e.g., records, unions, arrays, lists, functions, references)

Model statement (link, visibility, scope and life time).
General view of type checking.

</td><td>

For both a primitive and a compound type, informally describe the values that have that type [Familiarity]
For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity]
Describe examples of program errors detected by a type system [Familiarity]
For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage]
Give an example program that does not type-check in a particular language and yet would have no error if run [Familiarity]
Use types and type-error messages to write and debug programs [Usage]
Explain how typing rules define the set of operations that are legal for a type [Familiarity]
Write down the type rules governing the use of a particular compound type [Usage]
Explain why undecidability requires type systems to conservatively approximate program behavior [Familiarity]
Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]
Discuss the differences among generics, subtyping, and overloading [Familiarity]
Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Str13], [Dei17]</td></tr>
</table>

Unit 4: Fundamental Programming Concepts (6)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment] • Identify and describe uses of primitive data types [Familiarity] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Assessment] • Describe the concept of recursion and give examples of its use [Familiarity] • Identify the base case and the general case of a recursively-defined problem [Assessment]
Readings : [Str13], [Dei17]	

<table>
<tr><td colspan="2">Unit 5: Object-Oriented Programming (10)</td></tr>
<tr><td colspan="2">Competences Expected: a,b,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Object-oriented design

Decomposition into objects carrying state and having behavior
Class-hierarchy design for modeling

Object-oriented idioms for encapsulation

Privacy and visibility of class members
Interfaces revealing only method signatures
Abstract base classes

Definition of classes: fields, methods, and constructors
Subclasses, inheritance, and method overriding
Subtyping

Subtype polymorphism; implicit upcasts in typed languages
Notion of behavioral replacement: subtypes acting like supertypes
Relationship between subtyping and inheritance

Using collection classes, iterators, and other common library components
Dynamic dispatch: definition of method-call

</td><td>

Design and implement a class [Usage]
Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage]
Correctly reason about control flow in a program using dynamic dispatch [Usage]
Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Assessment]
Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Familiarity]
Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage]
Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Str13], [Dei17]</td></tr>
</table>

Unit 6: Algorithms and Design (3)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**

• Problem-solving strategies
  – Iterative and recursive mathematical functions
  – Iterative and recursive traversal of data structures
  – Divide-and-conquer strategies
• The role of algorithms in the problem-solving process
• Problem-solving strategies
  – Iterative and recursive mathematical functions
  – Iterative and recursive traversal of data structures
  – Divide-and-conquer strategies
• Fundamental design concepts and principles
  – Abstraction
  – Program decomposition
  – Encapsulation and information hiding
  – Separation of behavior and implementation
	• Discuss the importance of algorithms in the problem-solving process [Familiarity]
	• Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity]
	• Create algorithms for solving simple problems [Usage]
	• Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage]
	• Implement, test, and debug simple recursive functions and procedures [Usage]
	• Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment]
	• Implement a divide-and-conquer algorithm for solving a problem [Usage]
	• Apply the techniques of decomposition to break a program into smaller pieces [Usage]
	• Identify the data components and behaviors of multiple abstract data types [Usage]
	• Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage]
	• Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]

Readings : [Str13], [Dei17]	

Unit 7: Algorithmic Strategies (3)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Brute-force algorithms • Greedy algorithms • Divide-and-conquer • Recursive backtracking • Dynamic Programming	• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Familiarity] • Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment] • Use a divide-and-conquer algorithm to solve an appropriate problem [Usage] • Use recursive backtracking to solve a problem such as navigating a maze [Usage] • Use dynamic programming to solve an appropriate problem [Usage] • Determine an appropriate algorithmic approach to a problem [Assessment] • Describe various heuristic problem-solving methods [Familiarity]
Readings : [Str13], [Dei17]	

Unit 8: Basic Analysis (2)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Familiarity]
Readings : [Str13], [Dei17]	

Unit 9: Fundamental Data Structures and Algorithms (6)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)	• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Trace and/or implement a string-matching algorithm [Usage]
Readings : [Str13], [Dei17]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017. ISBN: 978-0201734843.

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.

Contents

1. COURSE

CS1D2. Discrete Structures II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D1. Discrete Structures I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

5. GOALS

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student applies efficient travel strategies to be able to search data in an optimal way.
- That the student uses the various counting techniques to solve computational problems.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Digital Logic and Data Representation (10)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Reticles: Types and properties. • Boolean algebras. • Boolean Functions and Expressions. • Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form. • Logical gates. • Circuit Minimization.	• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment]. • Explain the algebraic structures of reticulum and its types [Assessment]. • Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment]. • Explain the properties that satisfies a Boolean algebra [Assessment]. • Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment]. • Find the canonical forms of a Boolean function [Assessment]. • Represent a Boolean function as a Boolean circuit using logic gates [Assessment]. • Minimize a Boolean function. [Assessment].
Readings : [Ros07], [Gri03]	

Unit 2: Basics of Counting (40)	
Competences Expected: a	
Topics	**Learning Outcomes**

• Counting arguments
  – Set cardinality and counting
  – Sum and product rule
  – Inclusion-exclusion principle
  – Arithmetic and geometric progressions
• The pigeonhole principle
• Permutations and combinations
  – Basic definitions
  – Pascal's identity
  – The binomial theorem
• Solving recurrence relations
  – An example of a simple recurrence relation, such as Fibonacci numbers
  – Other examples, showing a variety of solutions
• Basic modular arithmetic
	• Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity]
	• Apply the pigeonhole principle in the context of a formal proof [Familiarity]
	• Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity]
	• Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity]
	• Solve a variety of basic recurrence relations [Familiarity]
	• Analyze a problem to determine underlying recurrence relations [Familiarity]
	• Perform computations involving modular arithmetic [Familiarity]

Readings : [Gri97]	

Unit 3: Graphs and Trees (40)	
Competences Expected: a	
Topics	**Learning Outcomes**

• Trees
  – Properties
  – Traversal strategies
• Undirected graphs
• Directed graphs
• Weighted graphs
• Spanning trees/forests
• Graph isomorphism
	• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity]
	• Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity]
	• Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity]
	• Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity]
	• Explain how to construct a spanning tree of a graph [Familiarity]
	• Determine if two graphs are isomorphic [Familiarity]

Readings : [Joh99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria.* Addison Wesley Iberoamericana, 1997.

[Joh99] Richard Johnsonbaugh. *Matemáticas Discretas.* Prentice Hall, México, 1999.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

MA101. Math II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The course is focused on developing skills in problem understanding, comprehension and application of mathematical models. To this end, an active and participatory methodology is developed with rational use of technology and collaborative work spaces. The sessions are theoretical and practical associated to contextualized situations that motivate the student to get involved in their understanding and solution. The course aims to address the following main topics which will be monitored every week, these topics are Vectors, Functions of Several Variables, Partial Derivatives, Double Integrals, Series and Ordinary Differential Equations of first order and second or more order

5. GOALS

- Ability to apply knowledge of mathematics.
- Ability to apply engineering knowledge.
- Ability to apply computer and mathematical knowledge

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Vectors (24)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Components, canonical, force or speed problems. • Angle between two vectors, calculate work for a constant force, moment of a force, volume. • Equation of line and plane, Drawing planes, Distance between points, planes and lines. • Calculate work by constant force, moment of a force, volume. • Drawing functions of two and three variables, contour lines.	• Express a vector by its components and use vector operations to interpret the results geometrically, using standard or canonical linear combinations of unit vectors. • Understand the three-dimensional rectangular coordinate system and analyze vectors in space; finding the angle between two vectors and the perpendicular vector between two vectors. • Apply knowledge about vector properties in physical and chemical properties. • Give a set of parametric equations for a line in space. • Give a linear equation to represent a plane in space, using it to draw the plane given by the linear equation. • Find the distances between points, planes and lines in space.
Readings : [Ste12], [Zil13]	

Unit 2: Derivatives and Integrals (12)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Interpreting the directional derivatives, error analysis, chain rule. • Directional derivative, gradient of a two-variable function, application. • Absolute and relative extremes / criteria of the second partial derivatives. • Areas, volumes and average values. • Double integrals using polar coordinates.	• Understand the notation for a multi-variable function, helping you to draw the graph in space. • Make contour plots of a two-variable function. • Find and use the partial derivatives of a function of two or more variables, to understand the concepts of increments and differentials. • Use a differential as an approximation and use the chain rule for multivariate functions. • Find and use the directional derivatives of a two-variable function, using it to find the gradient of a two-variable function. • Find absolute and relative ends of a two-variable function, using the criterion of the second partial derivatives. • Solve optimization problems with unrestricted and restricted multivariate functions, using the Lagrange multiplier method. • Evaluate and use an iterated integral to find the area of a flat region in Cartesian coordinates.
Readings : [Ste12], [Zil13]	

Unit 3: Series and Successions (24)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Succession - limit of a succession - recognition of patterns of a succession. • Infinite geometric series - integral and P series criteria. • Quotient criterion / Taylor and Maclaurin polynomials. • Taylor / Maclaurin series.	• Find the mass, center of mass and moments of inertia of a flat sheet using a double integral. • Determine if a succession converges or diverges, using limits and L'Hospital's rule. • Understand the definition of an infinite series using properties to find whether they are convergent or divergent. • Use criteria and properties of the infinite series to determine whether it is convergent or divergent. • Find polynomial approximations of functions using Taylor and Maclaurin polynomials to elementary functions. • Understand the definition of a power series to calculate the radius and range of convergence. • Find a Taylor or Maclaurin series for a function.
Readings : [Ste12], [Zil13]	

Unit 4: Differential Equations (30)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Definitions and terminologies / Problems with initial values. • Separable variable - Linear equations. • Linear Models of Growth (Population), Decay (Bacteria - Half-life - Mixtures - Newton's Law.) • Exact Equations - Solutions by substitution. • Nonlinear Models (Falling Chain - Logistic Population Growth - Leaking Cylindrical Tank - Inverted Cone, Solar Collector, Immigration Model. • Radioactive Series - Mixed - Mesh. • Nutrient concentration - Newton's Law. • Problems with initial values - homogeneous and non-homogeneous. • Annulator method - Cauchy Euler equation.	• Understand the definitions and terminology of differential equations with and without initial values • Explain 1st and 2nd order differential equation models. • Solve first-order differential equations by the separable variables method. • Solve the homogeneous and non-homogeneous first-order linear differential equations using the integral factor. • Solve exact first-order differential equations with and without initial values, using the integration factor. • Obtain the general solution of a homogeneous second order linear equation with constant coefficients. • Solve the Euler equation of second order, applying to analyze applications in mechanical vibrations and oscillations in electrical circuits.
Readings : [Ste12], [Zil13]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ste12] James Stewart. *Cálculo de varias variable Trascendentes tempranas*. Ed. by Cengage Learning Editores S.A. de C.V. 6th. 2012.

[Zil13] Dennis G. Zill. *Ecuaciones diferenciales con valores en la frontera*. Ed. by Cengage Learning Editores. 8th. 2013.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG106. Theater (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG101. Communication. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It helps students to identify themselves with the 'Academic Community' of the University, insofar as it provides them with natural channels of integration into their group and their Study Centre and allows them, from an alternative viewpoint, to visualise the inner worth of the people around them, while at the same time getting to know their own. It relates the university student, through experimentation, with a new language, a means of communication and expression that goes beyond the conceptualized verbal expression. It helps the student in his integral formation, developing in him corporal capacities. It stimulates positive attitudes, cognitive and affective skills. It enriches their sensitivity and awakens their solidarity. It disinhibits and socializes, relaxes and makes people happy, opening a path of knowledge of one's own being and the being of others.

5. GOALS

- To contribute to the personal and professional formation of the student, recognizing, valuing and developing his body language, integrating him to his group, strengthening his personal security, enriching his intuition, his imagination and creativity, motivating him to open paths of search of knowledge of himself and communication with others through his sensibility, exercises of introspection and new ways of expression.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (6)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• What is Art? An experiential and personal one. • The master key: creativity. • The importance of the theatre in personal and professional training. • Usefulness and focus of the theatrical art.	• Recognize the validity of Art and creativity in personal and social development [Usage]. • To relate the student to his group, valuing the importance of human communication and the social collective [Usage]. • Recognize basic notions of theater [Usage].
Readings : [Maj58], [Pav98]	

Unit 2: (6)	
Competences Expected: C17,C24	
Topics	**Learning Outcomes**
• I play, then I exist. • Child's play and dramatic play. • Group integration games and creativity games. • The theatrical sequence.a	• Recognize play as a fundamental tool of the theater. [Usage]. • Internalizing and revaluing play as creative learning. [Usage]. • To bring the student closer to the theatrical experience in a spontaneous and natural way. [Usage].
Readings : [Maj58], [Pav98]	

Unit 3: (9)	
Competences Expected: C17, C18, C24	
Topics	**Learning Outcomes**
• Awareness of the body. • Awareness of space • Time awareness • Creation of individual and collective sequences: Body, space and time • The dramatic use of the element: The theatrical game. • Theatrical presentations with the use of the element.	• Experimenting with new forms of expression and communication. [Usage]. • Know some mechanisms of control and body management. [Usage]. • To provide paths for the student to creatively develop his imagination, his ability to relate to and capture auditory, rhythmic and visual stimuli. [Usage]. • To know and develop the management of their own space and spatial relations . [Usage]. • Experiencing different emotional states and new collective climates. [Usage].
Readings : [Maj58], [Pav98]	

Unit 4: (12)	
Competences Expected: C18, C24	
Topics	**Learning Outcomes**
• Relaxation, concentration and breathing. • Disinhibition and interaction with the group. • Improvisation. • Balance, weight, time and rhythm. • Analysis of the movement. Types of movement. • The theatrical presence. • The dance, the theatrical choreography.	• Exercise in the management of non-verbal communication skills. [Usage]. • Practice games and body language exercises, individually and in groups. [Usage]. • To freely and creatively express their emotions and feelings and their vision of society through original representations in various languages. [Usage]. • Knowing the types of action. [Usage].
Readings : [Maj58], [Pav98]	

Unit 5: (3)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• The origin of the theatre, the Greek theatre and the Roman theatre. • The medieval theatre, the comedy of art. • From passion to reason: Romanticism and Enlightenment. • The realistic theatre, epic theatre. Brech and Stanislavski. • The theatre of the absurd, contemporary theatre and total theatre. • Theater in Peru: Yuyashkani, La Tarumba, pataclaun, others.	• To know the influence that society has exerted on the theatre and the response of this art to different moments in history. [Usage]. • To appreciate the value and contribution of the works of important playwrights. [Usage]. • Analyzing the social context of theatrical art. [Usage]. • Reflecting on Peruvian and Arequipa's Theatre. [Usage].
Readings : [Maj58], [Pav98]	

Unit 6: (12)	
Competences Expected: C17,C18, C24	
Topics	**Learning Outcomes**
• Theatrical appreciation. Expectation of one or more plays. • Theatrical space. • Construction of the character • Creation and staging of a play. • Public presentation of small plays using costumes, make-up, scenery, props and the dramatic use of the object.	• To use theatrical creation as a manifestation of one's own ideas and feelings before society. [Usage]. • To apply the techniques practiced and the knowledge learned in a concrete theatrical appreciation and/or expression that links the role of education. [Usage]. • Exchange experiences and make short presentations of theatrical exercises in groups, in front of an audience. [Usage].
Readings : [Maj58], [Pav98]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Maj58] Angel Majorana. *El arte de hablar en publico*. La España Moderna, 1958.

[Pav98] Patrice Pavis. *Diccionario del Teatro*. Edit. Piados BA, 1998.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS113. Computer Science II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS112. Computer Science I. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the third course in the sequence of introductory courses in computer science. This course is intended to cover Concepts indicated by the Computing Curriculum IEEE (c) -ACM 2001, under the functional-first approach. The object-oriented paradigm allows us to combat complexity by making models from abstractions of the problem elements and using techniques such as encapsulation, modularity, polymorphism and inheritance. The Dominion of these topics will enable participants to provide computational solutions to design problems simple of the real world.

5. GOALS

- Introduce the student in the fundaments of the paradigm of object orientation, allowing the assimilation of concepts necessary to develop an information system

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Programming Concepts (5)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Usage] • Identify and describe uses of primitive data types [Usage] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Usage] • Describe the concept of recursion and give examples of its use [Usage] • Identify the base case and the general case of a recursively-defined problem [Usage]
Readings : [**stroustrup2013**], [Van02], [LE13]	

Unit 2: Object-Oriented Programming (7)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Object-oriented design
  – Decomposition into objects carrying state and having behavior
  – Class-hierarchy design for modeling
• Definition of classes: fields, methods, and constructors
• Subclasses, inheritance, and method overriding
• Dynamic dispatch: definition of method-call
• Subtyping
  – Subtype polymorphism; implicit upcasts in typed languages
  – Notion of behavioral replacement: subtypes acting like supertypes
  – Relationship between subtyping and inheritance
• Object-oriented idioms for encapsulation
  – Privacy and visibility of class members
  – Interfaces revealing only method signatures
  – Abstract base classes
• Using collection classes, iterators, and other common library components
	• Design and implement a class [Usage]
	• Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage]
	• Correctly reason about control flow in a program using dynamic dispatch [Usage]
	• Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Usage]
	• Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Usage]
	• Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage]
	• Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]

Readings : [stroustrup2013]	

Unit 3: Algorithms and Design (5)	
Competences Expected: a,b,d	
Topics	**Learning Outcomes**

• The concept and properties of algorithms
  – Informal comparison of algorithm efficiency (e.g., operation counts)
• The role of algorithms in the problem-solving process
• Problem-solving strategies
  – Iterative and recursive mathematical functions
  – Iterative and recursive traversal of data structures
  – Divide-and-conquer strategies
• Fundamental design concepts and principles
  – Abstraction
  – Program decomposition
  – Encapsulation and information hiding
  – Separation of behaivor and implementation
	• Discuss the importance of algorithms in the problem-solving process [Usage]
	• Discuss how a problem may be solved by multiple algorithms, each with different properties [Usage]
	• Create algorithms for solving simple problems [Usage]
	• Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage]
	• Implement, test, and debug simple recursive functions and procedures [Usage]
	• Determine whether a recursive or iterative solution is most appropriate for a problem [Usage]
	• Implement a divide-and-conquer algorithm for solving a problem [Usage]
	• Apply the techniques of decomposition to break a program into smaller pieces [Usage]
	• Identify the data components and behaviors of multiple abstract data types [Usage]
	• Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage]
	• Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Usage]

Readings : [**stroustrup2013**], [**Weert16**], [LE13]	

Unit 4: Basic Analysis (3)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Empirical measurements of performance • Time and space trade-offs in algorithms • Big O notation: use • Little o, big omega and big theta notation • Recurrence relations • Analysis of iterative and recursive algorithms • Master Theorem and Recursion Trees	• Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm [Usage] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Usage] • Determine informally the time and space complexity of different algorithms [Usage] • State the formal definition of big O [Usage] • List and contrast standard complexity classes [Usage] • Perform empirical studies to validate hypotheses about runtime stemming from mathematical analysis Run algorithms on input of various sizes and compare performance [Usage] • Give examples that illustrate time-space trade-offs of algorithms [Usage] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Usage] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Usage] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Usage]
Readings : [stroustrup2013]	

<table>
<tr><td colspan="2">Unit 5: Basic Type Systems (5)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

A type as a set of values together with a set of operations

Primitive types (e.g., numbers, Booleans)
Compound types built from other types (e.g., records, unions, arrays, lists, functions, references)

Association of types to variables, arguments, results, and fields
Type safety and errors caused by using values inconsistently given their intended types
Goals and limitations of static typing

Eliminating some classes of errors without running the program
Undecidability means static analysis must conservatively approximate program behavior

Generic types (parametric polymorphism)

Definition
Use for generic libraries such as collections
Comparison with ad hoc polymorphism (overloading) and subtype polymorphism

Complementary benefits of static and dynamic typing

Errors early vs. errors late/avoided
Enforce invariants during code development and code maintenance vs. postpone typing decisions while prototyping and conveniently allow flexible coding patterns such as heterogeneous collections
Avoid misuse of code vs. allow more code reuse
Detect incomplete programs vs. allow incomplete programs to run

</td><td>

For both a primitive and a compound type, informally describe the values that have that type [Usage]
For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Usage]
Describe examples of program errors detected by a type system [Usage]
For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage]
Give an example program that does not type-check in a particular language and yet would have no error if run [Usage]
Use types and type-error messages to write and debug programs [Usage]
Explain how typing rules define the set of operations that are legal for a type [Usage]
Write down the type rules governing the use of a particular compound type [Usage]
Explain why undecidability requires type systems to conservatively approximate program behavior [Usage]
Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]
Discuss the differences among generics, subtyping, and overloading [Usage]
Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Usage]

</td></tr>
<tr><td colspan="2">Readings : [stroustrup2013]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 6: Fundamental Data Structures and Algorithms (3)</th></tr>
<tr><th colspan="2">Competences Expected: a,b,d</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max,
Sequential and binary search algorithms
Worst case quadratic sorting algorithms (selection, insertion)
Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)
Hash tables, including strategies for avoiding and resolving collisions
Binary search trees
Common operations on binary search trees such as select min, max, insert, delete, iterate over tree
Graphs and graph algorithms
Representations of graphs (e.g., adjacency list, adjacency matrix)
Depth- and breadth-first traversals
Heaps
Graphs and graph algorithms
Maximum and minimum cut problem
Local search
Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)

</td><td>

Implement basic numerical algorithms [Usage]
Implement simple search algorithms and explain the differences in their time complexities [Usage]
Be able to implement common quadratic and O(N log N) sorting algorithms [Usage]
Describe the implementation of hash tables, including collision avoidance and resolution [Usage]
Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Usage]
Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Usage]
Explain how tree balance affects the efficiency of various binary search tree operations [Usage]
Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage]
Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Usage]
Describe the heap property and the use of heaps as an implementation of priority queues [Usage]
Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage]
Trace and/or implement a string-matching algorithm [Usage]

</td></tr>
<tr><td colspan="2">Readings : [stroustrup2013], [PA18]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 7: Event-Driven and Reactive Programming (2)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Events and event handlers
Canonical uses such as GUIs, mobile devices, robots, servers
Using a reactive framework

Defining event handlers/listeners
Main event loop not under event-handler-writer's control

Externally-generated events and program-generated events
Separation of model, view, and controller

</td><td>

Write event handlers for use in reactive systems, such as GUIs [Usage]
Explain why an event-driven programming style is natural in domains where programs react to external events [Usage]
Describe an interactive system in terms of a model, a view, and a controller [Usage]

</td></tr>
<tr><td colspan="2">Readings : [stroustrup2013], [Wil11]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 8: Graphs and Trees (7)</td></tr>
<tr><td colspan="2">Competences Expected: a,b,d</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Trees

Properties
Traversal strategies

Undirected graphs
Directed graphs
Weighted graphs
Spanning trees/forests
Graph isomorphism

</td><td>

Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Usage]
Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Usage]
Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Usage]
Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Usage]
Explain how to construct a spanning tree of a graph [Usage]
Determine if two graphs are isomorphic [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Nak13]</td></tr>
</table>

Unit 9: Software Design (6)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures
• Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented
• Structural and behavioral models of software designs
• Design patterns
• Relationships between requirements and designs: transformation of models, design of contracts, invariants
• Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters)
• The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set)
• Refactoring designs using design patterns
• Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments
• Measurement and analysis of design quality
• Tradeoffs between different aspects of quality
• Application frameworks
• Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems
• Principles of secure design and coding
  – Principle of least privilege
  – Principle of fail-safe defaults
  – Principle of psychological acceptability
	• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Usage]
	• Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage]
	• Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage]
	• Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Usage]
	• For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage]
	• Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage]
	• Explain the relationships between the requirements for a software product and its design, using appropriate models [Usage]
	• For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Usage]
	• Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Usage]
	• Investigate the impact of software architectures selection on the design of a simple system [Usage]
	• Apply simple examples of patterns in a software design [Usage]
	• Describe a form of refactoring and discuss when it may be applicable [Usage]
	• Select suitable components for use in the design of a software product [Usage]
	• Explain how suitable components might need to be adapted for use in the design of a software product [Usage]
	• Design a contract for a typical small software component for use in a given system [Usage]
	• Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage]
	• Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-

Unit 10: Requirements Engineering (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing	• List the key components of a use case or similar description of some behavior that is required for a system [Usage] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Usage] • Interpret a given requirements model for a simple software system [Usage] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Usage] • List the key components of a data model (eg, class diagrams or ER diagrams) [Usage] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Usage] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Usage] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Usage] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Usage] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Usage] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Usage] • Create a prototype of a software system to mitigate risk in requirements [Usage] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Usage]
Readings : [**stroustrup2013**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[LE13] Stanley B. Lippman and Barbara E.Moo. *C++ Primer*. 5th. O'Reilly, 2013. ISBN: 9780133053043.

[Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platforml, 2013.

[PA18] Praseed Pai and Peter Abraham. *C++ Reactive Programming*. 1st. Packt, 2018.

[Van02] David Vandervoorde. *C++ Templates:The Complete Guide*. 1st. Addison-Wesley, 2002. ISBN: 978-0134448237.

[Wil11] Anthony Williams. *C++ Concurrency in Action*. 1st. Manning, 2011.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS221. Computer Systems Architecture (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A computer scientist must have a solid knowledge of the organization and design principles of diverse computer systems, by understanding the limitations of modern systems they could propose next-gen paradigms. This course teaches the basics and principles of Computer Architecture. This class addreses digital logic design, basics of Computer Architecture and processor design (Instruction Set architecture, microarchitecture, out-of-order execution, branch prediction), execution paradigms (superscalar, dataflow, VLIW, SIMD, GPUs, systolic, multithreading) and memory system organization.

5. GOALS

- Provide a first approach in Computer Architecture.
- Study the design and evolution of computer architectures, which lead to modern approaches and implementations in computing systems.
- Provide fine-grained details of computer hardware, and its relation with software execution.
- Implement a simple microprocessor using Verilog language.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Digital logic and digital systems (18)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Overview and history of computer architecture • Combinational and sequential logic/Field programmable gate arrays as a fundamental combinational + sequential logic building block • Abstraction models • Computer-aided design tools that process hardware and architectural representations • Register transfer notation/Hardware Description Language (Verilog/VHDL) • Physical constraints (gate delays, fan-in, fan-out, energy/power)	• Describe the progression of technology devices from vacuum tubes to VLSI, from mainframe computer architectures to the organization of warehouse-scale computers [Familiarity] • Comprehend the trend of modern computer architectures towards multi-core and that parallelism is inherent in all hardware systems [Usage] • Explain the implications of the "power wall" in terms of further processor performance improvements and the drive towards harnessing parallelism [Usage] • Articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits [Familiarity] • Design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), memory (register transfer-level) [Usage] • Use CAD tools for capture, synthesis, and simulation to evaluate simple building blocks (eg, arithmetic-logic unit, registers, movement between registers) of a simple computer design [Familiarity] • Evaluate the functional and timing diagram behavior of a simple processor implemented at the logic circuit level [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 2: Machine level representation of data (8)	
Competences Expected: g	
Topics	**Learning Outcomes**
• Bits, bytes, and words • Numeric data representation and number bases • Fixed- and floating-point systems • Signed and twos-complement representations • Representation of non-numeric data (character codes, graphical data) • Representation of registers and arrays	• Explain why everything is data, including instructions, in computers [Assessment] • Explain the reasons for using alternative formats to represent numerical data [Familiarity] • Describe how negative integers are stored in sign-magnitude and twos-complement representations [Usage] • Explain how fixed-length number representations affect accuracy and precision [Usage] • Describe the internal representation of non-numeric data, such as characters, strings, records, and arrays [Usage] • Convert numerical data from one format to another [Usage]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 3: Assembly level machine organization (8)	
Competences Expected: b,g	
Topics	**Learning Outcomes**
• Basic organization of the von Neumann machine • Control unit; instruction fetch, decode, and execution • Instruction sets and types (data manipulation, control, I/O) • Assembly/machine language programming • Instruction formats • Addressing modes • Subroutine call and return mechanisms • I/O and interrupts • Heap vs. Static vs. Stack vs. Code segments	• Explain the organization of the classical von Neumann machine and its major functional units [Familiarity] • Describe how an instruction is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution [Familiarity] • Describe instruction level parallelism and hazards, and how they are managed in typical processor pipelines [Familiarity] • Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler [Familiarity] • Demonstrate how to map between high-level language patterns into assembly/machine language notations [Usage] • Explain different instruction formats, such as addresses per instruction and variable length vs fixed length formats [Usage] • Explain how subroutine calls are handled at the assembly level [Usage] • Explain the basic concepts of interrupts and I/O operations [Familiarity] • Write simple assembly language program segments [Usage] • Show how fundamental high-level programming constructs are implemented at the machine-language level [Usage]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 4: Functional organization (8)	
Competences Expected: b,g	
Topics	**Learning Outcomes**
• Implementation of simple datapaths, including instruction pipelining, hazard detection and resolution • Control unit: microprogrammed • Instruction pipelining • Introduction to instruction-level parallelism (ILP)	• Compare alternative implementation of datapaths [Assessment] • Discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations [Familiarity] • Explain basic instruction level parallelism using pipelining and the major hazards that may occur [Usage] • Design and implement a complete processor, including datapath and control [Usage] • Determine, for a given processor and memory system implementation, the average cycles per instruction [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 5: Memory system organization and architecture (8)	
Competences Expected: b,g	
Topics	**Learning Outcomes**
• Storage systems and their technology • Memory hierarchy: importance of temporal and spatial locality • Main memory organization and operations • Latency, cycle time, bandwidth, and interleaving • Cache memories (address mapping, block size, replacement and store policy) • Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations • Virtual memory (page table, TLB) • Fault handling and reliability • Error coding, data compression, and data integrity	• Identify the main types of memory technology (eg, SRAM, DRAM, Flash, magnetic disk) and their relative cost and performance [Familiarity] • Explain the effect of memory latency on running time [Familiarity] • Describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency [Usage] • Describe the principles of memory management [Usage] • Explain the workings of a system with virtual memory management [Usage] • Compute Average Memory Access Time under a variety of cache and memory configurations and mixes of instruction and data references [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

<table>
<tr><td colspan="2">Unit 6: Interfacing and communication (8)</td></tr>
<tr><td colspan="2">Competences Expected: b,g,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O
Interrupt structures: vectored and prioritized, interrupt acknowledgment
External storage, physical organization, and drives
Buses: bus protocols, arbitration, direct-memory access (DMA)
Introduction to networks: communications networks as another layer of remote access
Multimedia support
RAID architectures

</td><td>

Explain how interrupts are used to implement I/O control and data transfers [Familiarity]
Identify various types of buses in a computer system [Familiarity]
Describe data access from a magnetic disk drive [Usage]
Compare common network organizations, such as ethernet/bus, ring, switched vs routed [Assessment]
Identify the cross-layer interfaces needed for multimedia access and presentation, from image fetch from remote storage, through transport over a communications network, to staging into local memory, and final presentation to a graphical display [Familiarity]
Describe the advantages and limitations of RAID architectures [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 7: Multiprocessing and alternative architectures (8)</td></tr>
<tr><td colspan="2">Competences Expected: i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Power Law
Example SIMD and MIMD instruction sets and architectures
Interconnection networks (hypercube, shuffle-exchange, mesh, crossbar)
Shared multiprocessor memory systems and memory consistency
Multiprocessor cache coherence

</td><td>

Discuss the concept of parallel processing beyond the classical von Neumann model [Assessment]
Describe alternative parallel architectures such as SIMD and MIMD [Familiarity]
Explain the concept of interconnection networks and characterize different approaches [Usage]
Discuss the special concerns that multiprocessing systems present with respect to memory management and describe how these are addressed [Familiarity]
Describe the differences between memory backplane, processor memory interconnect, and remote memory via networks, their implications for access latency and impact on program performance [Assessment]

</td></tr>
<tr><td colspan="2">Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]</td></tr>
</table>

Unit 8: Performance enhancements (8)	
Competences Expected: g,i	
Topics	**Learning Outcomes**
• Superscalar architecture • Branch prediction, Speculative execution, Out-of-order execution • Prefetching • Vector processors and GPUs • Hardware support for multithreading • Scalability • Alternative architectures, such as VLIW/EPIC, and Accelerators and other kinds of Special-Purpose Processors	• Describe superscalar architectures and their advantages [Familiarity] • Explain the concept of branch prediction and its utility [Usage] • Characterize the costs and benefits of prefetching [Assessment] • Explain speculative execution and identify the conditions that justify it [Assessment] • Discuss the performance advantages that multithreading offered in an architecture along with the factors that make it difficult to derive maximum benefits from this approach [Assessment] • Describe the relevance of scalability to performance [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann, 2012. ISBN: 978-0123944245.

[HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.

[JAs07] Peter J.Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann, 2007. ISBN: 978-0123695277.

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.

[PCh06] Pong P.Chu. *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience, 2006.

[PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.

[PP05] Yale N Patt and Sanjay J Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.

[Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS2B1. Platform Based Development (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS112. Computer Science I. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The world has changed due to the use of fabric and related technologies, rapid, timely and personalized access to the information, through web technology, ubiquitous and pervasive; they have changed the way we do things, how do we think? and how does the industry develop? Web technologies, ubiquitous and pervasive are based on the development of web services, web applications and mobile applications, which are necessary to understand the architecture, design, and implementation of web services, web applications and mobile applications.

5. GOALS

- That the student is able to design and implement services, web applications using tools and languages such as HTML, CSS, JavaScript (including AJAX), back-end scripting and a database, at an intermediate level.
- That the student is able to develop mobile applications, administration of web servers in a Unix system and an introduction to web security, at an intermediate level.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction (5)	
Competences Expected: g	
Topics	**Learning Outcomes**
• Overview of platforms (e.g., Web, Mobile, Game, Industrial) • Programming via platform-specific APIs • Overview of Platform Languages (e.g., Objective C, HTML5) • Programming under platform constraints	• Describe how platform-based development differs from general purpose programming [Familiarity] • List characteristics of platform languages [Familiarity] • Write and execute a simple platform-based program [Familiarity] • List the advantages and disadvantages of programming with platform constraints [Familiarity]
Readings : [fielding2000fielding], [grove2009web], [annuzzi2013introduction], [Cornez2015]	

Unit 2: Web Platforms (5)	
Competences Expected: c,g,i	
Topics	**Learning Outcomes**
• Web programming languages (e.g., HTML5, Java Script, PHP, CSS) • • Web Platform constraints: Client-Server, Stateless-Stateful, Cache, Uniform Interface, Layered System, Code on Demand, ReST. • Web platform constraints • Software as a Service (SaaS) • Web standards	• Design and Implement a simple web application [Familiarity] • Describe the constraints that the web puts on developers [Familiarity] • Compare and contrast web programming with general purpose programming [Familiarity] • Describe the differences between Software-as-a-Service and traditional software products [Familiarity] • Discuss how web standards impact software development [Familiarity] • Review an existing web application against a current web standard [Familiarity]
Readings : [fielding2000fielding]	

Unit 3: Desarrollo de servicios y aplicaciones web (25)	
Competences Expected: c,d,g,i	
Topics	**Learning Outcomes**
• Describe, identify and debug issues related to web application development • Design and development of interactive web applications using HTML5 and Python • Use MySQL for data management and manipulate MySQL with Python • Design and development of asynchronous web applications using Ajax techniques • Using dynamic client side Javascript scripting language and server side python scripting language with Ajax • Apply XML / JSON technologies for data management with Ajax • Use framework, services and Ajax web APIs and apply design patterns to web application development	• Server-side python scripting language: variables, data types, operations, strings, functions, control statements, arrays, files and directory access, maintain state. [Usage] • Web programming approach using embedded python. [Usage] • Accessing and Manipulating MySQL. [Usage] • The Ajax web application development approach. [Usage] • DOM and CSS used in JavaScript. [Usage] • Asynchronous Content Update Technologies. [Usage] • XMLHttpRequest objects use to communicate between clients and servers. [Usage] • XML and JSON. [Usage] • XSLT and XPath as mechanisms for transforming XML documents. [Usage] • Web services and APIs (especially Google Maps). [Usage] • Macros Ajax for the development of contemporary web applications. [Usage] • Design patterns used in web applications. [Usage]
Readings : [**freeman2011head**]	

Unit 4: Mobile Platforms (5)	
Competences Expected: c,d,g,i	
Topics	**Learning Outcomes**
• Mobile programming languages • Design Principles: Segregation of Interfaces, Single Responsability, Separation of concerns, Dependency Inversion. • Challenges with mobility and wireless communication • Location-aware applications • Performance / power tradeoffs • Mobile platform constraints • Emerging technologies	• Design and implement a mobile application for a given mobile platform [Familiarity] • Discuss the constraints that mobile platforms put on developers [Familiarity] • Discuss the performance vs power tradeoff [Familiarity] • Compare and Contrast mobile programming with general purpose programming [Familiarity]
Readings : [**martin2017clean**], [**annuzzi2013introduction**]	

Unit 5: Mobile Applications for Android Handheld Systems (25)	
Competences Expected: c,d,g,i	
Topics	**Learning Outcomes**
• The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps	• Students identify necessary software and install it on their personal computers. • Students perform various tasks to familiarize themselves with the Android platform and Environment for development. [Usage] • Students build applications that trace the lifecycle callback methods emitted by the Android platform and demonstrate the behavior of Android when device configuration changes (for example, when the device moves from vertical to horizontal and vice versa). [Usage] • Students build applications that require starting multiple activities through both standard and custom methods. [Usage] • Students build applications that require standard and custom permissions. [Usage] • Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usage] • Students construct a to-do list manager using the user interface elements discussed in class. The application allows users to create new items and to display them in a ListView. [Usage] • Students build an application that uses location information to collect latitude, length of places they visit. [Usage]
Readings : [**annuzzi2013introduction**], [**Cornez2015**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG203. Oratory (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG106. Theater. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In a competitive society such as ours, it is required that the person be an effective communicator and know how to use his or her potential to solve problems and face the challenges of the modern world within the work, intellectual and social activity. Having knowledge is not enough, the important thing is to know how to communicate it and to the extent that the person knows how to use his or her communicative faculties, what he or she has to do in his or her personal and professional development will derive in success or failure. Therefore it is necessary to achieve a good saying, to resort to knowledge, strategies and resources, which every speaker must have, to reach the interlocutor with clarity, precision and conviction.

5. GOALS

- At the end of the course, the student will be able to organize and assume the word from the speaker's perspective, in any situation, in a more correct, coherent and adequate way, through the use of knowledge and linguistic skills, seeking at all times their personal and social realization through their expression, based on truth and constant preparation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (3)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Oratory • The function of the word. • The process of communication. • Rational and emotional basis of public speaking – Oral expression in participation. • Sources of knowledge for public speaking: levels of general culture.	• Understanding: to interpret, exemplify and generalize the basis of oratory as a theoretical and practical foundation. [Usage].
Readings : [ME76], [Rod]	

Unit 2: (4)	
Competences Expected: C17	
Topics	**Learning Outcomes**

• Qualities of a good speaker.
• Rules for first speeches.
• The human body as an instrument of communication:
  – Body expression in speech
  – The voice in the speech.
• Speakers with history and their example.
	• Understanding: Interpreting, exemplifying and generalizing knowledge and skills of oral communication through the experience of great speakers and your own. [Usage].
	• Application: Implementing, using, choosing and performing the knowledge acquired to express yourself in public in an efficient, intelligent and pleasant way. [Usage].

Readings : [Rod]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[ME76] A. Monroe and D. Ehninger. *La comunicación oral.* Hispano Europea, 1976.

[Rod] María L. Rodríguez. *Cómo manejar la información en una presentación.*

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS210. Algorithms and Data Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS113. Computer Science II. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The theoretical foundation of all branches of computing rests on algorithms and data structures, this course will provide participants with an introduction to these topics, thus forming a basis that will serve for the following courses in the career.

5. GOALS

- Make the student understand the importance of algorithms for solving problems.
- Introduce the student to the field of application of data structures.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Graphs (12)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Graph Concept • Directed Graphs and Non-directed Graphs. • Using Graphs. • Measurement of efficiency ,in time and space. • Adjacency matrices. • Tag adjacent matrices. • Adjacency Lists. • Implementation of graphs using adjacency matrices. • Graph Implementation using adjacency lists • Insertion, search and deletion of nodes and edges. • Graph search algorithms.	• Acquire Dexterity to Perform Correct Implementation. [Usage] • Develop knowledge to decide when it is better to use one implementation technique than another. [Usage]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 2: Scatter Matrices (8)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Initial concepts. • Dense Matrices • Measurement of Efficiency in Time and Space • Static scatter vs. dynamic matrix creation. • Insert, search, and delete methods.	• Understand the use and implementation of scatter matrices.[Assessment]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 3: Balanced Trees (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• AVL Trees. • Measurement of Efficiency. • Simple and Composite Rotations • Insertion, deletion and search. • Trees B , B+ B* y Patricia.	• Understand the basic functions of these complex structures in order to acquire the capacity for their implementation. [Assessment]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms*. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.

[Fag+14] José Fager et al. *Estructura de datos*. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. 3rd. Addison-Wesley Professional, 1997.

[Knu98] Donald E. Knuth. *The art of computer programming, volume 3:Sorting and searching*. 2nd. Addison-Wesley Professional, 1998.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS211. Theory of Computation (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course emphasizes formal languages, computer models and computability, as well as the fundamentals of computational complexity and complete NP problems.

5. GOALS

- That the student learn the fundamental concepts of the theory of formal languages.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Basic Automata Computability and Complexity (20)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Finite-state machines • Regular expressions • The halting problem • Context-free grammars • Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack) • Turing machines, or an equivalent formal model of universal computation • Nondeterministic Turing machines • Chomsky hierarchy • The Church-Turing thesis • Computability • Rice's Theorem • Examples of uncomputable functions • Implications of uncomputability	• Discuss the concept of finite state machines [Assessment] • Design a deterministic finite state machine to accept a specified language [Assessment] • Generate a regular expression to represent a specified language [Assessment] • Explain why the halting problem has no algorithmic solution [Assessment] • Design a context-free grammar to represent a specified language [Assessment] • Define the classes P and NP [Assessment] • Explain the significance of NP-completeness [Assessment] • Explain the Church-Turing thesis and its significance [Familiarity] • Explain Rice's Theorem and its significance [Familiarity] • Provide examples of uncomputable functions [Familiarity] • Prove that a problem is uncomputable by reducing a classic known uncomputable problem to it [Familiarity]
Readings : [Mar10], [Lin11], [Sip12]	

Unit 2: Advanced Computational Complexity (20)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Review of the classes P and NP; introduce P-space and EXP • Polynomial hierarchy • NP-completeness (Cook's theorem) • Classic NP-complete problems • Reduction Techniques	• Define the classes P and NP (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Define the P-space class and its relation to the EXP class [Assessment] • Explain the significance of NP-completeness (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Provide examples of classic NP-complete problems [Assessment] • Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it [Assessment]
Readings : [Mar10], [Lin11], [Sip12], [HU13]	

<table>
<tr><td colspan="2">Unit 3: Advanced Automata Theory and Computability (20)</td></tr>
<tr><td colspan="2">Competences Expected: j</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Sets and languages

Regular languages
Review of deterministic finite automata (DFAs)
Nondeterministic finite automata (NFAs)
Equivalence of DFAs and NFAs
Review of regular expressions; their equivalence to finite automata
Closure properties
Proving languages non-regular, via the pumping lemma or alternative means

Context-free languages

Push-down automata (PDAs)
Relationship of PDAs and context-free grammars
Properties of context-free languages

</td><td>

Determine a language's place in the Chomsky hierarchy (regular, context-free, recursively enumerable) [Assessment]
Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs [Assessment]

</td></tr>
<tr><td colspan="2">Readings : [HU13], [Bro93]</td></tr>
</table>

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.

[HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.

[Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones and Bartlett Learning, 2011.

[Mar10] John Martin. *Introduction to Languages and the Theory of Computation.* 4th. McGraw-Hill, 2010.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation.* 3rd. Cengage Learning, 2012.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS271. Data Management (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS112. Computer Science I. (2^{nd} Sem) • CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques.It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

5. GOALS

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Database Systems (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Approaches to and evolution of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture and data independence • Use of a declarative query language • Systems supporting structured and/or stream content • Approaches for managing large volumes of data (e.g., noSQL database systems, use of MapReduce).	• Explain the characteristics that distinguish the database approach from the approach of programming with data files [Usage] • Describe the most common designs for core database system components including the query optimizer, query executor, storage manager, access methods, and transaction processor [Usage] • Cite the basic goals, functions, and models of database systems [Usage] • Describe the components of a database system and give examples of their use [Usage] • Identify major DBMS functions and describe their role in a database system [Usage] • Explain the concept of data independence and its importance in a database system [Usage] • Use a declarative query language to elicit information from a database [Usage] • Describe facilities that datatbases provide supporting structures and/or stream (sequence) data, eg, text [Usage] • Describe major approaches to storing and processing large volumes of data [Usage]
Readings : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unit 2: Data Modeling (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Data modeling • Conceptual models (e.g., entity-relationship, UML diagrams) • Spreadsheet models • Relational data models • Object-oriented models • Semi-structured data model (expressed using DTD or XML Schema, for example)	• Compare and contrast appropriate data models, including internal structures, for different types of data [Usage] • Describe concepts in modeling notation (eg, Entity-Relation Diagrams or UML) and how they would be used [Usage] • Define the fundamental terminology used in the relational data model [Usage] • Describe the basic principles of the relational data model [Usage] • Apply the modeling concepts and notation of the relational data model [Usage] • Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning [Usage] • Describe the differences between relational and semi-structured data models [Usage] • Give a semi-structured equivalent (eg, in DTD or XML Schema) for a given relational schema [Usage]
Readings : [SW04], [EN04], [KS02]	

Unit 3: Indexing (4)	
Competences Expected: b,d,i	
Topics	**Learning Outcomes**
• The impact of indices on query performance • The basic structure of an index • Keeping a buffer of data in memory • Creating indexes with SQL • Indexing text • Indexing the web (e.g., web crawling)	• Generate an index file for a collection of resources [Usage] • Explain the role of an inverted index in locating a document in a collection [Usage] • Explain how stemming and stop words affect indexing [Usage] • Identify appropriate indices for given relational schema and query set [Usage] • Estimate time to retrieve information, when indices are used compared to when they are not used [Usage] • Describe key challenges in web crawling, eg, detecting duplicate documents, determining the crawling frontier [Usage]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

<table>
<tr><th colspan="2">Unit 4: Relational Databases (14)</th></tr>
<tr><th colspan="2">Competences Expected: b,d,i</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

- Mapping conceptual schema to a relational schema
- Entity and referential integrity
- Relational algebra and relational calculus
- Relational Database design
- Functional dependency
- Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition
- Candidate keys, superkeys, and closure of a set of attributes
- Normal forms (BCNF)
- Multi-valued dependency (4NF)
- Join dependency (PJNF, 5NF)
- Representation theory

</td><td>

- Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage]
- Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage]
- Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage]
- Write queries in the relational algebra [Usage]
- Write queries in the tuple relational calculus [Usage]
- Determine the functional dependency between two or more attributes that are a subset of a relation [Usage]
- Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage]
- Compute the closure of a set of attributes under given functional dependencies [Usage]
- Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage]
- Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage]
- Describe the properties of BCNF, PJNF, 5NF [Usage]
- Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage]
- Describe what is a multi-valued dependency and what type of constraints it specifies [Usage]

</td></tr>
<tr><td colspan="2">Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]</td></tr>
</table>

Unit 5: Query Languages (12)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Overview of database languages • SQL (data definition, query formulation, update sub-language, constraints, integrity) • Selections • Projections • Select-project-join • Aggregates and group-by • Subqueries • QBE and 4th-generation environments • Different ways to invoke non-procedural queries in conventional languages • Introduction to other major query languages (e.g., XPATH, SPARQL) • Stored procedures	• Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints [Usage] • Use SQL to create tables and retrieve (SELECT) information from a database [Usage] • Evaluate a set of query processing strategies and select the optimal strategy [Usage] • Create a non-procedural query by filling in templates of relations to construct an example of the desired query result [Usage] • Embed object-oriented queries into a stand-alone language such as C++ or Java (eg, SELECT Col-Method() FROM Object) [Usage] • Write a stored procedure that deals with parameters and has some control flow, to provide a given functionality [Usage]
Readings : [Die01], [EN04], [Cel05], [KS02]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.

[Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.

[EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

[ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases*. 2nd. O'Reilly Media, 2015.

[KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.

[RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.

[RG03] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. 3rd. McGraw-Hill, 2003.

[SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.

[WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS2S1. Operating systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS221. Computer Systems Architecture. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

An Operating System (OS) manages the computing resources to complete the execution of multiple applications and their associated processes. This course teaches the design of modern operating systems; and introduces their fundamental concepts covering multiple-program execution, scheduling, memory management, file systems, and security. Also, the course includes programming activities on a minimal operating system to solve problems and extend its functionality. Notice that these activities require much time to complete. However, working on them provides valuable insight into operating systems.

5. GOALS

- Study the design of modern operating systems.

- Provide a practical experience by designing and implementing a minimal operating system.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Overview of Operating Systems (3)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>• Role and purpose of the operating system
• Functionality of a typical operating system
• Mechanisms to support client-server models.
• Design issues (efficiency, robustness, flexibility, portability, security, compatibility)
• Influences of security, networking, multimedia, windowing systems</td><td>• Explain the objectives and functions of modern operating systems [Familiarity]
• Analyze the tradeoffs inherent in operating system design [Assessment]
• Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve [Familiarity]
• Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems [Familiarity]
• Identify potential threats to operating systems and the security features design to guard against them [Familiarity]</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 2: Operating System Principles (6)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>• Operating Sistems Structure (monolithic, layered, modular, micro-kernel models)
• Abstractions, processes, and resources
• Concepts of application program interfaces (APIs)
• The evolution of hardware/software techniques and application needs
• Device organization
• Interrupts: methods and implementations
• Concept of user/system state and protection, transition to kernel mode</td><td>• Explain the concept of a logical layer [Familiarity]
• Explain the benefits of building abstract layers in hierarchical fashion [Familiarity]
• Describe the value of APIs and middleware [Familiarity]
• Describe how computing resources are used by application software and managed by system software [Familiarity]
• Contrast kernel and user mode in an operating system [Assessment]
• Discuss the advantages and disadvantages of using interrupt processing [Familiarity]
• Explain the use of a device list and driver I/O queue [Familiarity]</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

Unit 3: Concurrency (9)	
Competences Expected: b	
Topics	**Learning Outcomes**
• States diagrams • Structures (ready list, process control blocks, and so forth) • Dispatching and context switching • The role of interrupts • Managing atomic access to OS objects • Implementing synchronization primitives • Multiprocessor issues (spin-locks, reentrancy)	• Describe the need for concurrency within the framework of an operating system [Familiarity] • Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks [Usage] • Summarize the range of mechanisms that can be employed at the operating system level to realize concurrent systems and describe the benefits of each [Familiarity] • Explain the different states that a task may pass through and the data structures needed to support the management of many tasks [Familiarity] • Summarize techniques for achieving synchronization in an operating system (eg, describe how to implement a semaphore using OS primitives) [Familiarity] • Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system [Familiarity] • Create state and transition diagrams for simple problem domains [Usage]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 4: Scheduling and Dispatch (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Preemptive and non-preemptive scheduling • Schedulers and policies • Processes and threads • Deadlines and real-time issues	• Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes [Assessment] • Describe relationships between scheduling algorithms and application domains [Familiarity] • Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O [Familiarity] • Describe the difference between processes and threads [Familiarity] • Compare and contrast static and dynamic approaches to real-time scheduling [Assessment] • Discuss the need for preemption and deadline scheduling [Familiarity] • Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 5: Memory Management (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Review of physical memory and memory management hardware • Working sets and thrashing • Caching	• Explain memory hierarchy and cost-performance trade-offs [Familiarity] • Summarize the principles of virtual memory as applied to caching and paging [Familiarity] • Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Assessment] • Defend the different ways of allocating memory to tasks, citing the relative merits of each [Familiarity] • Describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction) [Familiarity] • Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<table>
<tr><td colspan="2">Unit 6: Security and Protection (6)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Overview of system security
Policy/mechanism separation
Security methods and devices
Protection, access control, and authentication
Backups

</td><td>

Articulate the need for protection and security in an OS [Familiarity]
Summarize the features and limitations of an operating system used to provide protection and security [Familiarity]
Explain the mechanisms available in an OS to control access to resources (cross reference IAS/Security Architecture and Systems Administration/Access Control/Configuring systems to operate securely as an IT system) [Familiarity]
Carry out simple system administration tasks according to a security policy, for example creating accounts, setting permissions, applying patches, and arranging for regular backups (cross reference IAS/Security Architecture and Systems Administration) [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 7: Virtual Machines (6)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Types of virtualization (including Hardware/Software, OS, Server, Service, Network)
Paging and virtual memory
Virtual file systems
Hypervisors
Portable virtualization; emulation vs. isolation
Cost of virtualization

</td><td>

Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity]
Differentiate emulation and isolation [Familiarity]
Evaluate virtualization trade-offs [Assessment]
Discuss hypervisors and the need for them in conjunction with different types of hypervisors [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 8: Device Management (6)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Characteristics of serial and parallel devices
Abstracting device differences
Buffering strategies
Direct memory access
Recovery from failures

</td><td>

Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate [Familiarity]
Identify the relationship between the physical hardware and the virtual devices maintained by the operating system [Familiarity]
Explain buffering and describe strategies for implementing it [Familiarity]
Differentiate the mechanisms used in interfacing a range of devices (including hand-held devices, networks, multimedia) to a computer and explain the implications of these for the design of an operating system [Familiarity]
Describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted [Familiarity]
Identify the requirements for failure recovery [Familiarity]
Implement a simple device driver for a range of possible devices [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 9: File Systems (6)</td></tr>
<tr><td colspan="2">Competences Expected: b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Files: data, metadata, operations, organization, buffering, sequential, nonsequential.
Directories: contents and structure.
File systems: partitioning, mount/unmount, virtual file systems.
Standard implementation techniques
Memory-mapped files
Special-purpose file systems.
Naming, searching, access, backups
Journaling and log-structured file systems

</td><td>

Describe the choices to be made in designing file systems [Familiarity]
Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each [Assessment]
Summarize how hardware developments have led to changes in the priorities for the design and the management of file systems [Familiarity]
Summarize the use of journaling and how log-structured file systems enhance fault tolerance [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]</td></tr>
</table>

Unit 10: Real Time and Embedded Systems (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Process and task scheduling • Memory/disk management requirements in a real-time environment • Failures, risks, and recovery. • Special concerns in real-time systems	• Describe what makes a system a real-time system [Familiarity] • Explain the presence of and describe the characteristics of latency in real-time systems [Familiarity] • Summarize special concerns that real-time systems present, including risk, and how these concerns are addressed [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 11: Fault Tolerance (3)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Fundamental concepts: reliable and available systems • Spatial and temporal redundancy • Methods used to implement fault tolerance • Examples of OS mechanisms for detection, recovery, restart to implement fault tolerance, use of these techniques for the OS's own services.	• Explain the relevance of the terms fault tolerance, reliability, and availability [Familiarity] • Outline the range of methods for implementing fault tolerance in an operating system [Familiarity] • Explain how an operating system can continue functioning after a fault occurs [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 12: System Performance Evaluation (3)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Why system performance needs to be evaluated? • What is to be evaluated? • Systems performance policies, e.g., caching, paging, scheduling, memory management, and security • Evaluation models: deterministic, analytic, simulation, or implementation-specific • How to collect evaluation data (profiling and tracing mechanisms)	• Describe the performance measurements used to determine how a system performs [Familiarity] • Explain the main evaluation models used to evaluate a system [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014. ISBN: 978-0985673529.

[Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.

[Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.

[Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 4/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.

[Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

MA203. Statistics and Probabilities (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It provides an introduction to probability theory and statistical inference with applications, needs in data analysis, design of random models and decision making.

5. GOALS

- An ability to design and conduct experiments, as well as to analyze and interpret data.
- An ability to identify, formulate, and solve real problems.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Variable Type (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Variable Type: Continuous, discrete	• Classify the relevant variables identified according to their type: continuous (interval and ratio), categorical (nominal, ordinal, dichotomous). • Identify the relevant variables of a system using a process approach.
Readings : [MRo14], [Men14]	

Unit 2: Descriptive Statistics (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Central Tendency (Mean, median, mode) • Dispersion (Range, standard deviation, quartile) • Graphics: histogram, boxplot, etc.: Communication ability.	• Use central tendency measures and dispersion measures to describe the data gathered. • Use graphics to communicate the characteristics of the data gathered.
Readings : [MRo14], [Men14]	

Unit 3: Inferential Statistics (6)	
Competences Expected: CS2	
Topics	**Learning Outcomes**
• Determination of the sample size • Confidence interval • Type I and type II error • Distribution type • Hypothesis test (t-student, means, proportions and ANOVA) • Relationships between variables: correlation, regression.	• Propose questions and hypotheses of interest. • Analyze the data gathered using different statistical tools to answer questions of interest. • Draw conclusions based on the analysis performed.
Readings : [MRo14], [Men14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Men14] Beaver Mendenhall. *Introducción a la probabilidad y estadística*. 13th. Cengage Learning, 2014.

[MRo14] Sheldon M.Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th. Academic Press, 2014.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG350. Leadership and Performance (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG203. Oratory. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

At present, the different organizations in the world demand from their members the exercise of leadership, this means assuming the challenges assigned with efficiency and eagerness to serve, being these demands necessary for the search of a more just and reconciled society. This challenge involves the need to form our students with a correct knowledge of themselves, with the capacity to judge reality objectively and to propose orientations that seek to positively modify the environment.

5. GOALS

- Develop knowledge, criteria, skills and attitudes to exercise leadership, in order to achieve effectiveness and service in the challenges assigned, thus contributing to building a better society.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (15)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• Leadership Theories: • Definition of Leadership. • Fundamentals of Leadership. • Integral Vision of the Human Being and Reasons for Action. • The practice of Virtue in the exercise of Leadership.	• Analyze and understand the theoretical bases of the Leadership exercise.[Familiarity] • Based on what is understood, assume the right attitude to put it into practice.[Familiarity] • Initiate a process of self-knowledge oriented to discover leadership traits in itself.[Familiarity]
Readings : [Pil02], [Man09], [Ale09], [D S], [Alf10]	

Unit 2: (15)	
Competences Expected: C17,C18,C24	
Topics	**Learning Outcomes**
• Competence Theory. • Recognition of Competencies. • Development Plan. • Mental Models. • Emotional Needs. • Emotional Profiles. • Motivational Vices.	• To know and develop leadership skills, focused on achieving effectiveness, without neglecting the duty of service to others.[Familiarity] • Recognize personal and group tendencies necessary for the exercise of Leadership.[Familiarity]
Readings : [Wil09], [Lui08], [Pil02], [Mar07]	

Unit 3: (18)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• The personal relationship with the team. • Integral leadership. • Accompaniment and discipleship. • Fundamentals of Unity.	• Develop teamwork skills[Familiarity]
Readings : [Gol12], [**CardonaP**], [**Hersey**], [Hun10], [Haw12], [**Ginebra**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ale09] Dianine-Havard Alexandre. *Perfil del Líder. Hacia un Liderazgo Virtuoso.* Ediciones Urano S.A, 2009.

[Alf10] Sonnenfeld Alfred. *Liderazgo Ético. La Sabiduría de decidir bien.* Ediciones Encuentro S.A Madrid y Nueva Revista de Madrid, 2010.

[D S] SJ Anthony. D' Souza. *Descubre tu Liderazgo.* Editorial Sal Terrae.

[Gol12] D. Goleman. *Inteligencia emocional.* Editorial Kairós., 2012.

[Haw12] Peter. Hawkins. *Coaching y liderazgo de equipos: coaching para un liderazgo con capacidad de transformación.* Ediciones Granica, 2012.

[Hun10] Phil. Hunsaker. *El nuevo arte de gestionar equipos: Un enfoque actual para guiar y motivar con éxito.* 2010.

[Lui08] Huete Luis. *Construye tu Sueño.* LID Editorial Empresarial, 2008.

[Man09] Ferreiro Pablo/Alcázar Manuel. *Gobierno de Personas en la Empresa.* Ediciones Universidad de Navarra EUNSA, 2009.

[Mar07] Chinchilla Nuria/Moragas Maruja. *Dueños de Nuestro Destino.* Editorial Ariel, 2007.

[Pil02] Cardona Pablo/García Lombardi Pilar. *Cómo desarrollar las Competencias de Liderazgo.* PAD Lima- Perú, Tercera Edición., 2002.

[Wil09] Cardona Pablo/ Helen Wilkinson. *Creciendo como Líder.* Ediciones Universidad de Navarra S.A (EUNSA), Primera Edición, 2009.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS212. Analysis and Design of Algorithms (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS210. Algorithms and Data Structures. (4^{th} Sem) • CS211. Theory of Computation. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

An algorithm is, essentially, a well-defined set of rules or instructions that allow solving a computational problem. The theoretical study of the performance of the algorithms and the resources used by them, usually time and space, allows us to evaluate if an algorithm is suitable for solving a specific problem, comparing it with other algorithms for the same problem or even delimiting the boundary between Viable and impossible. This matter is so important that even Donald E. Knuth defined Computer Science as the study of algorithms. This course will present the most common techniques used in the analysis and design of efficient algorithms, with the purpose of learning the fundamental principles of the design, implementation and analysis of algorithms for the solution of computational problems

5. GOALS

- Develop the ability to evaluate the complexity and quality of algorithms proposed for a given problem.
- Study the most representative, introductory algorithms of the most important classes of problems treated in computation.
- Develop the ability to solve algorithmic problems using the fundamental principles of algorithm design learneds.
- Be able to answer the following questions when a new algorithm is presented: How good is the performance ?, Is there a better way to solve the problem?

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Basic Analysis (10)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Asymptotic Notation • Analysis of iterative and recursive algorithms • Inductive proofs and correctness of algorithms • Master Theorem and Recursion Trees	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Assessment] • Determine informally the time and space complexity of different algorithms [Assessment] • List and contrast standard complexity classes [Assessment] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Assessment] • Analyze worst-case running times of algorithms using asymptotic analysis [Assessment] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Assessment] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Assessment] • Argue the correctness of algorithms using inductive proofs [Assessment]
Readings : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

Unit 2: Algorithmic Strategies (30)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Brute-force algorithms • Greedy algorithms • Divide-and-conquer • Dynamic Programming	• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Assessment] • Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment] • Use a divide-and-conquer algorithm to solve an appropriate problem [Assessment] • Use dynamic programming to solve an appropriate problem [Assessment] • Determine an appropriate algorithmic approach to a problem [Assessment]
Readings : [KT05], [DPV06], [RS09], [Als99]	

Unit 3: Fundamental Data Structures and Algorithms (6)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Graphs and graph algorithms
  – Maximum and minimum cut problem
  – Local search
• Cache oblivious algorithms
• Number theory and cryptography
	• Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity]
	• Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Assessment]
	• Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment]
	• Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Assessment]

Readings : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unit 4: Basic Automata Computability and Complexity (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack) • Reductions	• Define the classes P and NP [Familiarity] • Explain the significance of NP-completeness [Familiarity]
Readings : [KT05], [DPV06], [RS09]	

Unit 5: Advanced Data Structures Algorithms and Analysis (12)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Graphs (e.g, topological sort, finding strongly connected components, matching) • Randomized algorithms • Amortized analysis • Probabilistic analysis • Approximation Algorithms • Linear Programming	• Understand the mapping of real-world problems to algorithmic solutions (eg, as graph problems, linear programs, etc) [Familiarity] • Select and apply advanced analysis techniques (eg, amortized, probabilistic, etc) to algorithms [Usage]
Readings : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scientific, 1999. ISBN: 9789810237400.

[DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006. ISBN: 9780073523408.

[GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 2009. ISBN: 0470088540, 9780470088548.

[Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1*. Third Edition. Addison-Wesley, 1997. ISBN: 9780201896831. URL: `http://www-cs-faculty.stanford/~knuth/taocp.html`.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321295358.

[Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms*. Computer Science Press, 1992. ISBN: 9780716782438.

[RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844.

[SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013. ISBN: 9780133373486.

[SW11] R. Sedgewick and K. Wayne. *Algorithms*. Pearson Education, 2011. ISBN: 9780132762564.

[Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983. ISBN: 0-89871-187-8.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS2702. Data Management II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS271. Data Management. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• Storage and file structure • Indexed files • Hashed files • Signature files • B-trees • Files with dense index • Files with variable length records • Database efficiency and tuning	• Explain the concepts of records, record types, and files, as well as the different techniques for placing file records on disk [Usage] • Give examples of the application of primary, secondary, and clustering indexes [Usage] • Distinguish between a non-dense index and a dense index [Usage] • Implement dynamic multilevel indexes using B-trees [Usage] • Explain the theory and application of internal and external hashing techniques [Usage] • Use hashing to facilitate dynamic file expansion [Usage] • Describe the relationships among hashing, compression, and efficient database searches [Usage] • Evaluate costs and benefits of various hashing schemes [Usage] • Explain how physical database design affects database transaction efficiency [Usage]
Readings : [Bur04], [Cel05]	

Unit 2: Transaction Processing (12)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• Transactions • Failure and recovery • Concurrency control • Interaction of transaction management with storage, especially buffering	• Create a transaction by embedding SQL into an application program [Usage] • Explain the concept of implicit commits [Usage] • Describe the issues specific to efficient transaction execution [Usage] • Explain when and why rollback is needed and how logging assures proper rollback [Usage] • Explain the effect of different isolation levels on the concurrency control mechanisms [Usage] • Choose the proper isolation level for implementing a specified transaction protocol [Usage] • Identify appropriate transaction boundaries in application programs [Usage]
Readings : [Phi97], [Ram04]	

Unit 3: Information Storage and Retrieval (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• Documents, electronic publishing, markup, and markup languages • Tries, inverted files, PAT trees, signature files, indexing • Morphological analysis, stemming, phrases, stop lists • Term frequency distributions, uncertainty, fuzziness, weighting • Vector space, probabilistic, logical, and advanced models • Information needs, relevance, evaluation, effectiveness • Thesauri, ontologies, classification and categorization, metadata • Bibliographic information, bibliometrics, citations • Routing and (community) filtering • Multimedia search, information seeking behavior, user modeling, feedback • Information summarization and visualization • Faceted search (e.g., using citations, keywords, classification schemes) • Digital libraries • Digitization, storage, interchange, digital objects, composites, and packages • Metadata and cataloging • Naming, repositories, archives • Archiving and preservation, integrity • Spaces (conceptual, geographical, 2/3D, VR) • Architectures (agents, buses, wrappers/mediators), interoperability • Services (searching, linking, browsing, and so forth) • Intellectual property rights management, privacy, and protection (watermarking)	• Explain basic information storage and retrieval concepts [Usage] • Describe what issues are specific to efficient information retrieval [Usage] • Give applications of alternative search strategies and explain why the particular search strategy is appropriate for the application [Usage] • Design and implement a small to medium size information storage and retrieval system, or digital library [Usage] • Describe some of the technical solutions to the problems related to archiving and preserving information in a digital library [Usage]
Readings : [Pet98], [Ram04]	

Unit 4: Distributed Databases (36)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• Distributed DBMS – Distributed data storage – Distributed query processing – Distributed transaction model – Homogeneous and heterogeneous solutions – Client-server distributed databases • Parallel DBMS – Parallel DBMS architectures: shared memory, shared disk, shared nothing; – Speedup and scale-up, e.g., use of the MapReduce processing model – Data replication and weak consistency models	• Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design process [Usage] • Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer [Usage] • Explain how the two-phase commit protocol is used to deal with committing a transaction that accesses databases stored on multiple nodes [Usage] • Describe distributed concurrency control based on the distinguished copy techniques and the voting method [Usage] • Describe the three levels of software in the client-server model [Usage]
Readings : [M T99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[M T99] Patrick Valduriez M. Tamer Ozsu. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.

[Pet98] Julita Vassileva Peter Brusilovsky Alfred Kobsa. *Adaptive Hypertext and Hypermedia, First Edition*. Springer, 1998.

[Phi97] Eric Newcomer Philip A. Bernstein. *Principles of Transaction Processing, First Edition*. Morgan Kaufmann, 1997.

[Ram04] Shamkant B. Navathe Ramez Elmasri. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS291. Software Engineering I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS113. Computer Science II. (3^{rd} Sem) • CS271. Data Management. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The aim of developing software, except for extremely simple applications, requires the execution of a well-defined development process. Professionals in this area require a high degree of knowledge of the different models and development process, so that they are able to choose the most suitable for each development project. On the other hand, the development of medium and large-scale systems requires the use of pattern and component libraries and the mastery of techniques related to component-based design

5. GOALS

- Provide the student with a theoretical and practical framework for the development of software under quality standards.
- Familiarize the student with the software modeling and construction processes through the use of CASE tools.
- Students should be able to select architectures and ad-hoc technology platforms for deployment scenarios
- Applying component-based modeling to ensure variables such as quality, cost, and time-to-market in development processes.
- Provide students with best practices for software verification and validation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Requirements Engineering (18)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing	• List the key components of a use case or similar description of some behavior that is required for a system [Assessment] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Assessment] • Interpret a given requirements model for a simple software system [Assessment] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Assessment] • List the key components of a data model (eg, class diagrams or ER diagrams) [Assessment] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Assessment] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Assessment] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Assessment] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Assessment] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Assessment] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Assessment] • Create a prototype of a software system to mitigate risk in requirements [Assessment] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Assessment]
Readings : [ES14], [HF03]	

<table>
<tr><td colspan="2">Unit 2: Software Design (18)</td></tr>
<tr><td colspan="2">Competences Expected: i,k</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures
Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented
Structural and behavioral models of software designs
Design patterns
Relationships between requirements and designs: transformation of models, design of contracts, invariants
Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters)
The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set)
Refactoring designs using design patterns
Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments
Measurement and analysis of design quality
Tradeoffs between different aspects of quality
Application frameworks
Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems
Principles of secure design and coding

Principle of least privilege
Principle of fail-safe defaults
Principle of psychological acceptability

</td><td>

Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Familiarity]
Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage]
Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage]
Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Familiarity]
For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage]
Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage]
Explain the relationships between the requirements for a software product and its design, using appropriate models [Assessment]
For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Familiarity]
Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Familiarity]
Investigate the impact of software architectures selection on the design of a simple system [Assessment]
Apply simple examples of patterns in a software design [Usage]
Describe a form of refactoring and discuss when it may be applicable [Familiarity]
Select suitable components for use in the design of a software product [Usage]
Explain how suitable components might need to be adapted for use in the design of a software product [Familiarity]
Design a contract for a typical small software component for use in a given system [Usage]
Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage]
Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-

</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 3: Software Construction (24)</td></tr>
<tr><td colspan="2">Competences Expected: i,k</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Coding practices: techniques, idioms/patterns, mechanisms for building quality programs

Defensive coding practices
Secure coding practices
Using exception handling mechanisms to make programs more robust, fault-tolerant

Coding standards
Integration strategies
Development context: “green field” vs. existing code base

Change impact analysis
Change actualization

Potential security problems in programs

Buffer and other types of overflows
Race conditions
Improper initialization, including choice of privileges
Checking input
Assuming success and correctness
Validating assumptions

</td><td>

Describe techniques, coding idioms and mechanisms for implementing designs to achieve desired properties such as reliability, efficiency, and robustness [Assessment]
Build robust code using exception handling mechanisms [Assessment]
Describe secure coding and defensive coding practices [Assessment]
Select and use a defined coding standard in a small software project [Assessment]
Compare and contrast integration strategies including top-down, bottom-up, and sandwich integration [Assessment]
Describe the process of analyzing and implementing changes to code base developed for a specific project [Assessment]
Describe the process of analyzing and implementing changes to a large existing code base [Assessment]
Rewrite a simple program to remove common vulnerabilities, such as buffer overflows, integer overflows and race conditions [Assessment]
Write a software component that performs some non-trivial task and is resilient to input and run-time errors [Assessment]

</td></tr>
<tr><td colspan="2">Readings : [ES14], [HF03]</td></tr>
</table>

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[ES14] Bert Bates Eric Freeman Elisabeth Robson and Kathy Sierra. *Head First Design Patterns*. 2nd. O'Reilly Media, Inc, July 2014.

[HF03] Brian Lyons Hans-Erik Eriksson Magnus Penker and Davis Fado. *UML 2 Toolkit*. 2nd. Wiley, Oct. 2003.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS342. Compilers (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS211. Theory of Computation. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

That the student knows and understands the concepts and fundamental principles of the theory of compilation to realize the construction of a compiler

5. GOALS

- Know the basic techniques used during the process of intermediate generation, optimization and code generation.

- Learning to implement small compilers.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Program Representation (5)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Programs that take (other) programs as input such as interpreters, compilers, type-checkers, documentation generators • Abstract syntax trees; contrast with concrete syntax • Data structures to represent code for execution, translation, or transmission • Just-in-time compilation and dynamic recompilation • Other common features of virtual machines, such as class loading, threads, and security.	• Explain how programs that process other programs treat the other programs as their input data [Familiarity] • Describe an abstract syntax tree for a small language [Familiarity] • Describe the benefits of having program representations other than strings of source code [Familiarity] • Write a program to process some representation of code for some purpose, such as an interpreter, an expression optimizer, or a documentation generator [Familiarity] • Explain the use of metadata in run-time representations of objects and activation records, such as class pointers, array lengths, return addresses, and frame pointers [Familiarity] • Discuss advantages, disadvantages, and difficulties of just-in-time and dynamic recompilation [Familiarity] • Identify the services provided by modern language run-time systems [Familiarity]
Readings : [Lou04b]	

Unit 2: Language Translation and Execution (10)	
Competences Expected: a,b,j	
Topics	**Learning Outcomes**

• Interpretation vs. compilation to native code vs. compilation to portable intermediate representation
• Language translation pipeline: parsing, optional type-checking, translation, linking, execution
  – Execution as native code or within a virtual machine
  – Alternatives like dynamic loading and dynamic (or “just-in-time”) code generation
• Run-time representation of core language constructs such as objects (method tables) and first-class functions (closures)
• Run-time layout of memory: call-stack, heap, static data
  – Implementing loops, recursion, and tail calls
• Memory management
  – Manual memory management: allocating, deallocating, and reusing heap memory
  – Automated memory management: garbage collection as an automated technique using the notion of reachability
	• Distinguish a language definition (what constructs mean) from a particular language implementation (compiler vs interpreter, run-time representation of data objects, etc) [Assessment]
	• Distinguish syntax and parsing from semantics and evaluation [Assessment]
	• Sketch a low-level run-time representation of core language constructs, such as objects or closures [Assessment]
	• Explain how programming language implementations typically organize memory into global data, text, heap, and stack sections and how features such as recursion and memory management map to this memory model [Assessment]
	• Identify and fix memory leaks and dangling-pointer dereferences [Assessment]
	• Discuss the benefits and limitations of garbage collection, including the notion of reachability [Assessment]

Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 3: Syntax Analysis (10)	
Competences Expected: a,b,j	
Topics	**Learning Outcomes**
• Scanning (lexical analysis) using regular expressions • Parsing strategies including top-down (e.g., recursive descent, Earley parsing, or LL) and bottom-up (e.g., backtracking or LR) techniques; role of context-free grammars • Generating scanners and parsers from declarative specifications	• Use formal grammars to specify the syntax of languages [Assessment] • Use declarative tools to generate parsers and scanners [Assessment] • Identify key issues in syntax definitions: ambiguity, associativity, precedence [Assessment]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 4: Compiler Semantic Analysis (15)	
Competences Expected: a,b,j	
Topics	**Learning Outcomes**
• High-level program representations such as abstract syntax trees • Scope and binding resolution • Type checking • Declarative specifications such as attribute grammars	• Implement context-sensitive, source-level static analyses such as type-checkers or resolving identifiers to identify their binding occurrences [Assessment] • Describe semantic analyses using an attribute grammar [Assessment]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 5: Code Generation (20)	
Competences Expected: a,b,j	
Topics	**Learning Outcomes**
• Procedure calls and method dispatching • Separate compilation; linking • Instruction selection • Instruction scheduling • Register allocation • Peephole optimization	• Identify all essential steps for automatically converting source code into assembly or other low-level languages [Assessment] • Generate the low-level code for calling functions/methods in modern languages [Assessment] • Discuss why separate compilation requires uniform calling conventions [Assessment] • Discuss why separate compilation limits optimization because of unknown effects of calls [Assessment] • Discuss opportunities for optimization introduced by naive translation and approaches for achieving optimization, such as instruction selection, instruction scheduling, register allocation, and peephole optimization [Assessment]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Aho+11] Alfred Aho et al. *Compilers Principles Techniques And Tools*. 2nd. ISBN:10-970-26-1133-4. Pearson, 2011.

[App02] A. W. Appel. *Modern compiler implementation in Java*. 2.a edición. Cambridge University Press, 2002.

[Lou04a] Kenneth C. Louden. *Compiler Construction: Principles and Practice*. Thomson, 2004.

[Lou04b] Kenneth C. Louden. *Lenguajes de Programacion*. Thomson, 2004.

[TS98] Bernard Teufel and Stephanie Schmidt. *Fundamentos de Compiladores*. Addison Wesley Iberoamericana, 1998.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CB111. Computational Physics (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The course develops the knowledge and skills to recognize, evaluate and apply the effects of physical phenomena related to mechanics in the field of engineering. In industry in general, the control of processes, the operation of machines, their maintenance, etc., are always governed by some kind of physical manifestation. Because of this, it is important for the student to understand the foundations of physical phenomena, the laws that govern them, their manifestation and the way to detect them. This course will allow the student to understand and identify the physical phenomena related to mechanics in order to control their effects on some technical process.

5. GOALS

- Ability to apply science knowledge.
- Ability to design and conduct experiments.
- Ability to apply computer and mathematical knowledge.
- Ability to develop research principles at an international level.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Work, Energy and Power (6)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Definition of work and the relationship between net work and kinetic energy. • Power and Efficiency.	• Determine the variables that affect the opposition to translation and opposition to rotation (moment of inertia) and calculate the kinetic energy of translation and rotation. • Calculate the work of a force, apply the Net Work and Energy Theorem to a real life system, and determine the power and efficiency.
Readings : [Hug13], [Hew07]	

Unit 2: Kinematics (6)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Spatial and temporal reference systems. • Average speed, average acceleration, linear and angular. • Position, velocity and acceleration vectors, linear and angular • Relationship between linear and angular kinematics.	• Understand the concepts of spatial and temporal reference system kinematics and trajectory and determine position, velocity, linear and angular acceleration, according to a physical or graphical context. • Decompose the linear acceleration, according to a coordinate system, in order to describe the position and in radial and tangential acceleration. • It determines position, speed and acceleration, using differential and integral calculus.
Readings : [Hug13], [Hew07]	

Unit 3: Newton's three laws (6)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Newton's 3 laws and their application to particles. • Moment of a force. • Rotation of a rigid body.	• To propose the rotation and translation equations for a solid and apply Newton's laws. • Analyze the characteristics of the friction force. Calculate the net radial force and the net centripetal force. • Calculate the center of mass and analyze the relationship between the variables of net force, time and speed change.
Readings : [Hug13], [Hew07]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Hew07] Paul Hewitt. *Física conceptual*. 10th. Pearson Educación, 2007.

[Hug13] Roger A. Freedman Hugh D. Young. *Física universitaria*. 13th. Pearson, 2013.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS261. Intelligent Systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA203. Statistics and Probabilities. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.

- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Fundamental Issues (2)</td></tr>
<tr><td colspan="2">Competences Expected: a</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Overview of AI problems, examples of successful recent AI applications
What is intelligent behavior?

The Turing test
Rational versus non-rational reasoning

Problem characteristics

Fully versus partially observable
Single versus multi-agent
Deterministic versus stochastic
Static versus dynamic
Discrete versus continuous

Nature of agents

Autonomous versus semi-autonomous
Reflexive, goal-based, and utility-based
The importance of perception and environmental interactions

Philosophical and ethical issues.

</td><td>

Describe Turing test and the “Chinese Room” thought experiment [Usage]
Determing the characteristics of a given problem that an intelligent systems must solve [Usage]

</td></tr>
<tr><td colspan="2">Readings : [De 06], [Pon+14]</td></tr>
</table>

Unit 2: Agents (2)	
Competences Expected: a	
Topics	**Learning Outcomes**

• Definitions of agents
• Agent architectures (e.g., reactive, layered, cognitive)
• Agent theory
• Rationality, game theory
  – Decision-theoretic agents
  – Markov decision processes (MDP)
• Software agents, personal assistants, and information access
  – Collaborative agents
  – Information-gathering agents
  – Believable agents (synthetic characters, modeling emotions in agents)
• Learning agents
• Multi-agent systems
  – Collaborating agents
  – Agent teams
  – Competitive agents (e.g., auctions, voting)
  – Swarm systems and biologically inspired models
	• List the defining characteristics of an intelligent agent [Usage]
	• Characterize and contrast the standard agent architectures [Usage]
	• Describe the applications of agent theory to domains such as software agents, personal assistants, and believable agents [Usage]
	• Describe the primary paradigms used by learning agents [Usage]
	• Demonstrate using appropriate examples how multi-agent systems support agent interaction [Usage]

Readings : [Nil01], [RN03], [Pon+14]	

Unit 3: Basic Search Strategies (2)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Problem spaces (states, goals and operators), problem solving by search • Factored representation (factoring state into variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heuristics and informed search (hill-climbing, generic best-first, A*) • Space and time efficiency of search • Two-player games (introduction to minimax search) • Constraint satisfaction (backtracking and local search methods)	• Formulate an efficient problem space for a problem expressed in natural language (eg, English) in terms of initial and goal states, and operators [Usage] • Describe the role of heuristics and describe the trade-offs among completeness, optimality, time complexity, and space complexity [Usage] • Describe the problem of combinatorial explosion of search space and its consequences [Usage] • Compare and contrast basic search issues with game playing issues [Usage]
Readings : [Nil01], [Pon+14]	

<table>
<tr><th colspan="2">Unit 4: Advanced Search (18)</th></tr>
<tr><th colspan="2">Competences Expected: a,j</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

Stochastic search

Simulated annealing
Genetic algorithms
Monte-Carlo tree search

Constructing search trees, dynamic search space, combinatorial explosion of search space
Implementation of A* search, beam search
Minimax search, alpha-beta pruning
Expectimax search (MDP-solving) and chance nodes

</td><td>

Design and implement a genetic algorithm solution to a problem [Usage]
Design and implement a simulated annealing schedule to avoid local minima in a problem [Usage]
Design and implement A*,beam search to solve a problem [Usage]
Apply minimax search with alpha-beta pruning to prune search space in a two-player game [Usage]
Compare and contrast genetic algorithms with classic search techniques [Usage]
Compare and contrast various heuristic searches vis-a-vis applicability to a given problem [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Gol89], [Nil01], [RN03], [Pon+14]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 5: Reasoning Under Uncertainty (18)</th></tr>
<tr><th colspan="2">Competences Expected: a,j</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

Review of basic probability
Random variables and probability distributions

Axioms of probability
Probabilistic inference
Bayes' Rule

Conditional Independence
Knowledge representations

Bayesian Networks

Exact inference and its complexity
Randomized sampling (Monte Carlo) methods (e.g. Gibbs sampling)

Markov Networks
Relational probability models
Hidden Markov Models

</td><td>

Apply Bayes' rule to determine the probability of a hypothesis given evidence [Usage]
Explain how conditional independence assertions allow for greater efficiency of probabilistic systems [Usage]
Identify examples of knowledge representations for reasoning under uncertainty [Usage]
State the complexity of exact inference Identify methods for approximate inference [Usage]

</td></tr>
<tr><td colspan="2">Readings : [KF09], [RN03]</td></tr>
</table>

Unit 6: Basic Machine Learning (4)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Definition and examples of broad variety of machine learning tasks, including classification • Inductive learning • Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees • The over-fitting problem • Measuring classifier accuracy	• List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised [Usage] • Identify examples of classification tasks, including the available input features and output to be predicted [Usage] • Explain the difference between inductive and deductive learning [Usage] • Describe over-fitting in the context of a problem [Usage] • Apply the simple statistical learning algorithm such as Naive Bayesian Classifier to a classification task and measure the classifier's accuracy [Usage]
Readings : [Mit98], [RN03], [Pon+14]	

Unit 7: Advanced Machine Learning (20)	
Competences Expected: a,j	
Topics	**Learning Outcomes**

• Definition and examples of broad variety of machine learning tasks
• General statistical-based learning, parameter estimation (maximum likelihood)
• Inductive logic programming (ILP)
• Supervised learning
  – Learning decision trees
  – Learning neural networks
  – Support vector machines (SVMs)
• Unsupervised Learning and clustering
  – EM
  – K-means
  – Self-organizing maps
• Semi-supervised learning
• Learning graphical models
• Performance evaluation (such as cross-validation, area under ROC curve)
• Application of Machine Learning algorithms to Data Mining (cross-reference IM/Data Mining)
	• Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised [Usage]
	• Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning [Usage]
	• Determine which of the three learning styles is appropriate to a particular problem domain [Usage]
	• Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks [Usage]
	• Evaluate the performance of a simple learning system on a real-world dataset [Usage]
	• Characterize the state of the art in learning theory, including its achievements and its shortcomings [Usage]
	• Explain the problem of overfitting, along with techniques for detecting and managing the problem [Usage]

Readings : [RN03], [KF09], [Mur12]	

Unit 8: Natural Language Processing (12)	
Competences Expected: a,j	
Topics	**Learning Outcomes**

• Deterministic and stochastic grammars
• Parsing algorithms
  – CFGs and chart parsers (e.g. CYK)
  – Probabilistic CFGs and weighted CYK
• Representing meaning / Semantics
  – Logic-based knowledge representations
  – Semantic roles
  – Temporal representations
  – Beliefs, desires, and intentions
• Corpus-based methods
• N-grams and HMMs
• Smoothing and backoff
• Examples of use: POS tagging and morphology
• Information retrieval
  – Vector space model
    ∗ TF & IDF
  – Precision and recall
• Information extraction
• Language translation
• Text classification, categorization
  – Bag of words model
	• Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each [Usage]
	• Simulate, apply, or implement classic and stochastic algorithms for parsing natural language [Usage]
	• Identify the challenges of representing meaning [Usage]
	• List the advantages of using standard corpora Identify examples of current corpora for a variety of NLP tasks [Usage]
	• Identify techniques for information retrieval, language translation, and text classification [Usage]

Readings : [Nil01], [RN03], [Pon+14]	

Unit 9: Perception and Computer Vision (12)	
Competences Expected: a,j	
Topics	**Learning Outcomes**

• Computer vision
  – Image acquisition, representation, processing and properties
  – Shape representation, object recognition and segmentation
  – Motion analysis
• Modularity in recognition
• Approaches to pattern recognition
  – Classification algorithms and measures of classification quality
  – Statistical techniques
	• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Usage]
	• List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses [Usage]
	• Implement 2d object recognition based on contour-and/or region-based shape representations [Usage]
	• Provide at least two examples of a transformation of a data source from one sensory domain to another, eg, tactile data interpreted as single-band 2d images [Usage]
	• Implement a feature-extraction algorithm on real data, eg, an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal [Usage]
	• Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification [Usage]
	• Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above [Usage]

Readings : [Nil01], [RN03], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.

[Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN: 0262013193.

[Mit98] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998.

[Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.

[Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión*. McGraw-Hill, 2001.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

[RN03] Stuart Russell and Peter Norvig. *Inteligencia Artifical: Un enfoque moderno*. Prentice Hall, 2003.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS292. Software Engineering II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS291. Software Engineering I. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The topics of this course extend the ideas of software design and development from the introduction sequence to programming to encompass the problems encountered in large-scale projects. It is a broader and more complete view of Software Engineering appreciated from a Project point of view.

5. GOALS

- Enable students to be part of and define software development teams facing real-world problems.
- familiarize the students with the process of administering a software project in such a way as to be able to create, improve and use tools and metrics that allow them to carry out the estimation and monitoring of a software project
- Create, evaluate and execute a test plan for medium-sized code segments, Distinguish between different types of tests, lay the foundation for creating, improve test procedures and tools for these purposes
- Select with justification an appropriate set of tools to support the development of a range of software products.
- Create, improve and use existing patterns for software maintenance. Disclose features and design patterns for software reuse.
- Identify and discuss different specialized systems, create, improve and use specialized standards for the design, implementation, maintenance and testing of specialized systems.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Tools and Environments (12)</td></tr>
<tr><td colspan="2">Competences Expected: c,f,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Software configuration management and version control
• Release management
• Requierements analysis and desing modeling tools
• Testing tools including static and dynamic analysis tools
• Programming enviroments that automate parts of program construction pocesses (e.g., automated builds)
  – Continuous integration
• Tool integration concepts and mechanisms
</td><td>
• Software configuration management and version control [Usage]
• Release management [Usage]
• Requierements analysis and desing modeling tools [Usage]
• Testing tools including static and dynamic analysis tools [Usage]
• Programming enviroments that automate parts of program construction pocesses (e.g., automated builds)
  – Continuous integration
[Usage]
• Tool integration concepts and mechanisms [Usage]
</td></tr>
<tr><td colspan="2">Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 2: Software Verification and Validation (12)</td></tr>
<tr><td colspan="2">Competences Expected: c,f,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

- Verification and validation concepts
- Inspections, reviews, audits
- Testing types, including human computer interface, usability, reliability, security, conformance to specification
- Testing fundamentals
  - Unit, integration, validation, and system testing
  - Test plan creation and test case generation
  - Black-box and white-box testing techniques
  - Regression testing and test automation
- Defect tracking
- Limitations of testing in particular domains, such as parallel or safety-critical systems
- Static approaches and dynamic approaches to verification
- Test-driven development
- Validation planning; documentation for validation
- Object-oriented testing; systems testing
- Verification and validation of non-code artifacts (documentation, help files, training materials)
- Fault logging, fault tracking and technical support for such activities
- Fault estimation and testing termination including defect seeding

</td><td>

- Distinguish between program validation and verification [Usage]
- Describe the role that tools can play in the validation of software [Usage]
- Undertake, as part of a team activity, an inspection of a medium-size code segment [Usage]
- Describe and distinguish among the different types and levels of testing (unit, integration, systems, and acceptance) [Usage]
- Describe techniques for identifying significant test cases for integration, regression and system testing [Usage]
- Create and document a set of tests for a medium-size code segment [Usage]
- Describe how to select good regression tests and automate them [Usage]
- Use a defect tracking tool to manage software defects in a small software project [Usage]
- Discuss the limitations of testing in a particular domain [Usage]
- Evaluate a test suite for a medium-size code segment [Usage]
- Compare static and dynamic approaches to verification [Usage]
- Identify the fundamental principles of test-driven development methods and explain the role of automated testing in these methods [Usage]
- Discuss the issues involving the testing of object-oriented software [Usage]
- Describe techniques for the verification and validation of non-code artifacts [Usage]
- Describe approaches for fault estimation [Usage]
- Estimate the number of faults in a small software application based on fault density and fault seeding [Usage]
- Conduct an inspection or review of software source code for a small or medium sized software project [Usage]

</td></tr>
<tr><td colspan="2">Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]</td></tr>
</table>

Unit 3: Software Evolution (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**

• Software development in the context of large, pre-existing code bases
  – Software change
  – Concerns and concernlocation
  – Refactoring
• Software evolution
• Characteristics of maintainable software
• Reengineering systems
• Software reuse
  – Code segments
  – Libraries and frameworks
  – Components
  – Product lines
	• Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Usage]
	• Estimate the impact of a change request to an existing product of medium size [Usage]
	• Use refactoring in the process of modifying a software component [Usage]
	• Discuss the challenges of evolving systems in a changing environment [Usage]
	• Outline the process of regression testing and its role in release management [Usage]
	• Discuss the advantages and disadvantages of different types of software reuse [Usage]

Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 4: Software Project Management (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**

• Team participation
  – Team processes including responsabilities for task, meeting structure, and work schedule
  – Roles and responsabilities in a software team
  – Team conflict resolution
  – Risks associated with virtual teams (communication, perception, structure)
• Effort estimation (at the personal level)
• Risk
  – The role of risk in the lifecycle
  – Risk categories including security, safety, market, financial, technology, people, quality, structure and process
• Team management
  – Team organization and decision-making
  – Role identification and assigment
  – Individual and team performance assessment
• Project management
  – Scheduling and tracking
  – Project management tools
  – Cost/benefit analysis
• Software measurement and estimation techniques
• Software quality assurance and the role of measurements
• Risk
  – Risk identification and management
  – Risk analysis and evaluation
  – Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking)
  – Risk planning
• System-wide approach to risk including hazards associated with tools
	• Discuss common behaviors that contribute to the effective functioning of a team [Usage]
	• Create and follow an agenda for a team meeting [Usage]
	• Identify and justify necessary roles in a software development team [Usage]
	• Understand the sources, hazards, and potential benefits of team conflict [Usage]
	• Apply a conflict resolution strategy in a team setting [Usage]
	• Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage]
	• List several examples of software risks [Usage]
	• Describe the impact of risk in a software development lifecycle [Usage]
	• Describe different categories of risk in software systems [Usage]
	• Demonstrate through involvement in a team project the central elements of team building and team management [Usage]

Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Amb01] Vincenzo Ambriola. *Software Process Technology.* Springer, July 2001.

[Blu92] Bruce I. Blum. *Software Engineering: A Holistic View.* 7th. Oxford University Press US, May 1992.

[Con00] R Conradi. *Software Process Technology.* Springer, Mar. 2000.

[Key04] Jessica Keyes. *Software Configuration Management.* CRC Press, Feb. 2004.

[Mon96] Carlo Montangero. *Software Process Technology.* Springer, Sept. 1996.

[Oqu03] Flavio Oquendo. *Software Process Technology.* Springer, Sept. 2003.

[Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach.* 6th. McGraw-Hill, Mar. 2004.

[PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing.* Marcel Dekker, Jan. 2001.

[Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering.* McGraw-Hill, Jan. 2004.

[WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process.* Prentice Hall, Aug. 2002.

[WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications.* CRC Press, Apr. 2000.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS311. Competitive Programming (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Competitive Programming combines problem-solving challenges with the fun of competing with others. It teaches participants to think faster and develop problem-solving skills that are in high demand in the industry. This course will teach you to solve algorithmic problems quickly by combining theory of algorithms and data structures with practice solving problems.

5. GOALS

- That the student uses techniques of data structures and complex algorithms..
- That the student apply the concepts learned for the application on a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Introduction to Competetive Programming • Computacional model • Runtime and space complexity • Recurrence and recursion • Divide and conquer	• Identify and learn how to use the resources in the Random Access Machine (RAM) computacional model. [Usage] • Compute the runtime and space complexity for written algorithms. [Usage] • Compute the recurrence relations for recursive algorithms. [Usage] • Solve problems related to searching and sorting. [Usage] • Learning to select the right algorithms for divide-and-conquer problems. [Usage] • Design new algorithms for real-world problem solving.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 2: Data structure (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Arrays and strings problems • Linked lists problems • Stacks and queues problems • Trees problems • Hash tables problems • Heaps problems	• Recognize different data structures, their complexities, uses and restrictions.[Usage] • Identify the type of data structure appropriate to the resolution of the problem. [Usage] • Recognize types of problems associated with operations on data structures such as searching, inserting, deleting and updating.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 3: Algorithmic Design Paradigms (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Brute force • Divide and conquer • Backtracking • Greedy • Dynamic Programming	• Learning the different algorithhmic design paradigms.[Usage] • Learning to select the right algorithms for different problems applying different algorithhmic design paradigms.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 4: Graphs (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Graphs transversal • Graphs aplications • Shortest path • Networks and flows	• Identify problems classified as graph problems. [Usage] • Learn how to select the right algorithms for network problems (transversal, MST, shortest-path, network and flows). [Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 5: Advanced topics (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Number theory • Probabilities and combinations • String algorithms (tries, string hashing, z-algorithm) • Geometric algorithms	• Learning to select the right algorithms for problems in number theory and mathematics as they are important in competitive programming. [Usage] • Learning to select the right algorithms for problems about probabilities and combinations, strings and computational geometry. [Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 6: Domain specific problems (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Latency and throughput • Parallelism • Networks • Storage • High availability • Caching • Proxies • Load balancers • Key-value stores • Replicating and sharing • Leader election • Rate limiting • Logging and monitoring	• Learning to design systems for different domain-specific problems by applying knowledge about networks, distributed computing, high availability, storage and system architecture.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. ISBN: 9781479274833. URL: https://books.google.com.pe/books?id=y6FLBQAAQBAJ.

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.

[Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.

[Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.

[Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.

[Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003. ISBN: 978-0387001630.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS312. Advanced Data Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Algorithms and data structures are a fundamental part of computer science that allow us to organize information more efficiently, so it is important for every professional in the area to have a solid background in this regard.

In the course of advanced data structures our goal is for the student to know and analyze complex structures, such as Multidimensional Access Methods, Spatio-Temporal Access Methods and Metric Access Methods, Compact Data Structures, etc.

5. GOALS

- That the student understands, designs, implements, applies and Propose innovative data structures to solve problems related to the handling of multidimensional data, retrieval of information by similarity, search engines and other computational problems.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Basic techniques to implement data structuras (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Structured Programming • Object-oriented programming • Abstract Data Types • Independence of the user programming language of the structure • Platform Independence • Concurrency control • Data Protection • Encapsulation levels (struct, class, namespace, etc)	• That the student understands the basic differences that involve the different techniques of implementation of data structures[Usage] • That the student analyze the advantages and disadvantages of each of the existing techniques[Usage]
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94], [Bjö18], [Dav18]	

Unit 2: Multidimensional access methods (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Access Methods for Point Data • Access Methods for non-point data • Problems with dimension enhancement	• That the student understands to know and implement some Access Methods for multidimensional data and temporal space[Usage] • That the student understands the potential of these Access Methods in the future of commercial databases[Usage]
Readings : [Sam06], [Gü98]	

Unit 3: Metric access methods (20)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Metric Access Methods for discrete distances • Metric Access Methods for Continuous Distances	• That the student understands to know and implement some methods of metric access[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by similarity[Usage]
Readings : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unit 4: Approximate access methods (20)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Space Filling Curves • Locality Sensitive Hashing	• That the student understands to know and implement some approximate access methods[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by Similarity in environments where Scalability is a very important factor [Usage]
Readings : [Sam06], [PI06], [Zez+07]	

Unit 5: Seminars (8)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Access Methods Temporary Space • Generic Data Structures	• That the student can discuss the latest advances in access methods for different domains of knowledge [Usage]
Readings : [Sam06], [Nav16], [Chá+01]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bjö18] Stefan Björnander. *C++17 By Example: Practical projects to get you up and running with C++17*. Packt Publishing, Feb. 2018.

[Chá+01] E. Chávez et al. “Proximity Searching in Metric Spaces”. In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 273–321.

[Cua+04] Ernesto Cuadros-Vargas et al. “Implementing data structures: An incremental approach”. http://socios.spc.org.pe/ecuadros/cursos/pdfs/. 2004.

[Dav18] Doug Gregor David Vandevoorde Nicolai M. Josuttis. *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Sept. 2018. URL: `http://informit.com/aw`.

[Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.

[Gü98] Volker Gaede and Oliver ünther. "Multidimensional Access Methods". In: *ACM Computing Surveys* 30.2 (1998), pp. 170–231.

[Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms*. 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.

[Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching*. 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.

[Nav16] Gonzalo Navarro. *Compact Data Structures*. Cambridge University Press, 2016. ISBN: 978-1107152380.

[PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Illustrated. Elsevier/Morgan Kaufmann, Aug. 2006. ISBN: 9780123694461. URL: `http://books.google.com.pe/books?id=vO-NRRKHG84C`.

[Tra+00] C. Traina Jr et al. "Slim-Trees: High Performance Metric Trees Minimizing Overlap between Nodes". In: *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology*. Vol. 1777. Lecture Notes in Computer Science. Konstanz, Germany: Springer, Mar. 2000, pp. 51–65.

[Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach*. 1st. ISBN-10: 0387291466. Springer, Nov. 2007.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS393. Information systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS291. Software Engineering I. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Analyze techniques for the correct implementation of scalable, robust, reliable and efficient information systems in organizations.

5. GOALS

- Implement correctly (scalable, robust, reliable and efficient) Information Systems in organizations.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction (15)	
Competences Expected: c,i	
Topics	**Learning Outcomes**
• Introduction to information management. • Software for information management. • Technology for information management.	• Correctly apply technology for information management [Assessment]
Readings : [Som17], [PM15], [LL17]	

Unit 2: Strategy (15)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• Strategy for information management. • Strategy for knowledge management • Strategy for information system.	• Apply and evaluate correctly management strategies [Assessment]
Readings : [Som17], [PM15]	

Unit 3: Implementation (15)	
Competences Expected: c,i,k	
Topics	**Learning Outcomes**
• Management Information Systems Development. • Change management • Information Architecture	• Implement and correctly evaluate implementation strategies [Assessment]
Readings : [Som17], [PM15]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[LL17] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. 15th. Pearson, Mar. 2017.

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

MA307. Mathematics applied to computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• MA101. Math II. (2^{nd} Sem) • CB111. Computational Physics. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course is important because it develops topics of Linear Algebra and Ordinary Differential Equations useful in all areas of computer science where one works with linear systems and dynamic systems.

5. GOALS

- That the student has the mathematical basis for the modeling of linear systems and dynamic systems needed in the area of Computer Graphics and Artificial Intelligence.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (0)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Vector spaces. • Independence, base and dimension. • Dimensions and orthogonality of the four subspaces. • Approximations by least squares. • Projections • Orthogonal and Gram-Schmidt bases	• Identify spaces generated by linearly independent vectors. [Usage] • Build orthogonal vector arrays. [Usage] • Approximate functions by trigonometric polynomials. [Usage]
Readings : [Str03], [Apó73]	

Unit 2: (0)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Concept of linear transformation. • Matrix of a linear transformation. • Change of base. • Diagonalization and pseudo-inversion	• Determining the core and image of a transformation. [Usage] • Building the matrix of a transformation. [Usage] • Determine the base change matrix. [Usage]
Readings : [Str03], [Apó73]	

Unit 3: (0)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Diagonalization of a matrix. • Symmetrical matrices. • Positive defined matrices. • Similar matrices. • The decomposition of singular value.	• Finding the diagonal representation of a matrix. [Usage] • Determining similarity between matrices. [Usage] • Reducing a real quadratic shape to a diagonal. [Usage]
Readings : [Str03], [Apó73]	

Unit 4: (0)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Exponential of a matrix. • Theorems of existence and uniqueness for homogeneous linear systems with constant coefficients. • Non-homogeneous linear systems with constant coefficients.	• Finding the overall solution for a non-homogeneous linear system. [Usage] • Solving problems involving systems of differential equations. [Usage]
Readings : [Zil02], [Apó73]	

Unit 5: (0)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Dynamic systems. • The fundamental theorem. • Existence and uniqueness. • The flow of a differential equation.	• Discuss the existence and uniqueness of a differential equation. [Usage] • Analyze the continuity of solutions. [Usage] • Study the prolongation of a solution. [Usage]
Readings : [HS74]	

Unit 6: (0)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Stability. • Liapunov features. • Gradient systems.	• Analyze the stability of a solution. [Usage] • Finding Liapunov's function for balance points. [Usage] • Drawing the phase portrait a gradient flow. [Usage]
Readings : [Zil02], [HS74]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Apó73] Tom M Apóstol. *Calculus Vol II.* Editorial Reverté, 1973.

[HS74] Morris W. Hirsh and Stephen Smale. *Differential Equatons, Dynamical Systems, and Linear Álgebra.* Academia Press, 1974.

[Str03] Gilbert Strang. *Introduction to Linear Algebra, 3^a edición.* Wellesley-Cambridge Press, 2003.

[Zil02] Dennis G. Zill. *Ecuaciones Diferenciales con Problemas de Valores en la Frontera.* Thomson Learning, 2002. ISBN: 970-686-133-5.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS231. Networking and Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS2S1. Operating systems . (4^{th} Sem) • CS2S1. Operating systems . (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The ever-growing development of communication and information technologies means that there is a marked tendency to establish more computer networks that allow better information management..

In this second course, participants will be introduced to the problems of communication between computers, through the study and implementation of communication protocols such as TCP / IP and the implementation of software on these protocols

5. GOALS

- That the student implements and / or modifies a data communication protocols.
- That the student master the data transmission techniques used by the existing network protocols.
- That the student knows the latest trends in networks that are being applied on the Internet.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction (5)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Organization of the Internet (Internet Service Providers, Content Providers, etc.) • Switching techniques (e.g., circuit, packet) • Physical pieces of a network, including hosts, routers, switches, ISPs, wireless, LAN, access point, and firewalls • Layering principles (encapsulation, multiplexing) • Roles of the different layers (application, transport, network, datalink, physical)	• Articulate the organization of the Internet [Familiarity] • List and define the appropriate network terminology [Familiarity] • Describe the layered structure of a typical networked architecture [Familiarity] • Identify the different types of complexity in a network (edges, core, etc) [Familiarity]
Readings : [KR13]	

Unit 2: Networked Applications (5)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.) • Distributed applications (client/server, peer-to-peer, cloud, etc.) • HTTP as an application layer protocol • Multiplexing with TCP and UDP • Socket APIs	• List the differences and the relations between names and addresses in a network [Familiarity] • Define the principles behind naming schemes and resource location [Familiarity] • Implement a simple client-server socket-based application [Usage]
Readings : [KR13]	

Unit 3: Reliable Data Delivery (10)	
Competences Expected: C6,b,c,i	
Topics	**Learning Outcomes**
• Error control (retransmission techniques, timers) • Flow control (acknowledgements, sliding window) • Performance issues (pipelining) • TCP	• Describe the operation of reliable delivery protocols [Familiarity] • List the factors that affect the performance of reliable delivery protocols [Familiarity] • Design and implement a simple reliable protocol [Usage]
Readings : [KR13]	

Unit 4: Routing and Forwarding (12)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Routing versus forwarding • Static routing • Internet Protocol (IP) • Scalability issues (hierarchical addressing)	• Describe the organization of the network layer [Familiarity] • Describe how packets are forwarded in an IP network [Familiarity] • List the scalability benefits of hierarchical addressing [Familiarity]
Readings : [KR13]	

Unit 5: Local Area Networks (10)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Multiple Access Problem • Common approaches to multiple access (exponential-backoff, time division multiplexing, etc) • Local Area Networks • Ethernet • Switching	• Describe how frames are forwarded in an Ethernet network [Familiarity] • Describe the interrelations between IP and Ethernet [Familiarity] • Describe the steps used in one common approach to the multiple access problem [Familiarity]
Readings : [KR13]	

Unit 6: Resource Allocation (12)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Need for resource allocation • Fixed allocation (TDM, FDM, WDM) versus dynamic allocation • End-to-end versus network assisted approaches • Fairness • Principles of congestion control • Approaches to Congestion (e.g., Content Distribution Networks)	• Describe how resources can be allocated in a network [Familiarity] • Describe the congestion problem in a large network [Familiarity] • Compare and contrast fixed and dynamic allocation techniques [Familiarity] • Compare and contrast current approaches to congestion [Familiarity]
Readings : [KR13]	

Unit 7: Mobility (5)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Principles of cellular networks • 802.11 networks • Issues in supporting mobile nodes (home agents)	• Describe the organization of a wireless network [Familiarity] • Describe how wireless networks support mobile users [Familiarity]
Readings : [KR13], [Cha16]	

Unit 8: Social Networking (5)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Social networks overview • Example social network platforms • Structure of social network graphs • Social network analysis	• Discuss the key principles (such as membership, trust) of social networking [Familiarity] • Describe how existing social networks operate [Familiarity] • Construct a social network graph from network data [Usage] • Analyze a social network to determine who the key people are [Usage] • Evaluate a given interpretation of a social network question with associated data [Familiarity]
Readings : [KR13], [Kad11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016. ISBN: 978-0134463056.

[Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011. ISBN: 978-0195379471.

[KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013. ISBN: 978-0133594140.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS231. Networking and Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS2S1. Operating systems . (4^{th} Sem) • CS2S1. Operating systems . (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The ever-growing development of communication and information technologies means that there is a marked tendency to establish more computer networks that allow better information management..

In this second course, participants will be introduced to the problems of communication between computers, through the study and implementation of communication protocols such as TCP / IP and the implementation of software on these protocols

5. GOALS

- That the student implements and / or modifies a data communication protocols.
- That the student master the data transmission techniques used by the existing network protocols.
- That the student knows the latest trends in networks that are being applied on the Internet.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Introduction (5)</td></tr>
<tr><td colspan="2">Competences Expected: b,c</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Organization of the Internet (Internet Service Providers, Content Providers, etc.)
• Switching techniques (e.g., circuit, packet)
• Physical pieces of a network, including hosts, routers, switches, ISPs, wireless, LAN, access point, and firewalls
• Layering principles (encapsulation, multiplexing)
• Roles of the different layers (application, transport, network, datalink, physical)
</td><td>
• Articulate the organization of the Internet [Familiarity]
• List and define the appropriate network terminology [Familiarity]
• Describe the layered structure of a typical networked architecture [Familiarity]
• Identify the different types of complexity in a network (edges, core, etc) [Familiarity]
</td></tr>
<tr><td colspan="2">Readings : [KR13]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 2: Networked Applications (5)</td></tr>
<tr><td colspan="2">Competences Expected: b,c,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.)
• Distributed applications (client/server, peer-to-peer, cloud, etc.)
• HTTP as an application layer protocol
• Multiplexing with TCP and UDP
• Socket APIs
</td><td>
• List the differences and the relations between names and addresses in a network [Familiarity]
• Define the principles behind naming schemes and resource location [Familiarity]
• Implement a simple client-server socket-based application [Usage]
</td></tr>
<tr><td colspan="2">Readings : [KR13]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 3: Reliable Data Delivery (10)</td></tr>
<tr><td colspan="2">Competences Expected: C6,b,c,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Error control (retransmission techniques, timers)
• Flow control (acknowledgements, sliding window)
• Performance issues (pipelining)
• TCP
</td><td>
• Describe the operation of reliable delivery protocols [Familiarity]
• List the factors that affect the performance of reliable delivery protocols [Familiarity]
• Design and implement a simple reliable protocol [Usage]
</td></tr>
<tr><td colspan="2">Readings : [KR13]</td></tr>
</table>

Unit 4: Routing and Forwarding (12)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Routing versus forwarding • Static routing • Internet Protocol (IP) • Scalability issues (hierarchical addressing)	• Describe the organization of the network layer [Familiarity] • Describe how packets are forwarded in an IP network [Familiarity] • List the scalability benefits of hierarchical addressing [Familiarity]
Readings : [KR13]	

Unit 5: Local Area Networks (10)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Multiple Access Problem • Common approaches to multiple access (exponential-backoff, time division multiplexing, etc) • Local Area Networks • Ethernet • Switching	• Describe how frames are forwarded in an Ethernet network [Familiarity] • Describe the interrelations between IP and Ethernet [Familiarity] • Describe the steps used in one common approach to the multiple access problem [Familiarity]
Readings : [KR13]	

Unit 6: Resource Allocation (12)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Need for resource allocation • Fixed allocation (TDM, FDM, WDM) versus dynamic allocation • End-to-end versus network assisted approaches • Fairness • Principles of congestion control • Approaches to Congestion (e.g., Content Distribution Networks)	• Describe how resources can be allocated in a network [Familiarity] • Describe the congestion problem in a large network [Familiarity] • Compare and contrast fixed and dynamic allocation techniques [Familiarity] • Compare and contrast current approaches to congestion [Familiarity]
Readings : [KR13]	

Unit 7: Mobility (5)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Principles of cellular networks • 802.11 networks • Issues in supporting mobile nodes (home agents)	• Describe the organization of a wireless network [Familiarity] • Describe how wireless networks support mobile users [Familiarity]
Readings : [KR13], [Cha16]	

Unit 8: Social Networking (5)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Social networks overview • Example social network platforms • Structure of social network graphs • Social network analysis	• Discuss the key principles (such as membership, trust) of social networking [Familiarity] • Describe how existing social networks operate [Familiarity] • Construct a social network graph from network data [Usage] • Analyze a social network to determine who the key people are [Usage] • Evaluate a given interpretation of a social network question with associated data [Familiarity]
Readings : [KR13], [Kad11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016. ISBN: 978-0134463056.

[Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011. ISBN: 978-0195379471.

[KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013. ISBN: 978-0133594140.

Contents

1. COURSE

CS2H1. User Experience (UX) (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS393. Information systems. (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Language has been one of the most significant creations of humanity. From body language and gesture, through verbal and written communication, to iconic symbolic codes and others, it has made possible complex interactions Among humans and facilitated considerably the communication of information. With the invention of automatic and semi-automatic devices, including computers, The need for languages or interfaces to be able to interact with them, has gained great importance. The utility of the software, coupled with user satisfaction and increased productivity, depends on the effectiveness of the User-Computer Interface. So much so, that often the interface is the most important factor in the success and failure of any computer system. The design and implementation of appropriate Human-Computer Interfaces, which in addition to complying with the technical requirements and the transactional logic of the application, consider the subtle psychological implications, sciences and user facilities, It consumes a good part of the life cycle of a software project, and requires specialized skills, both for the construction of the same, and for the performance of usability tests.

5. GOALS

- Know and apply criteria of usability and accessibility to the design and construction of human-computer interfaces, always looking for technology to adapt to people and not people to technology.

- That the student has a vision focused on the user experience by applying appropriate conceptual and technological approaches.

- Understand how emerging technology makes possible new styles of interaction.

- Determine the basic requirements at the interface level, hardware and software for the construction of immersive environments.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Foundations (8)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Contexts for HCI (anything with a user interface, e.g., webpage, business applications, mobile applications, and games) • Usability heuristics and the principles of usability testing • Processes for user-centered development, e.g., early focus on users, empirical testing, iterative design • Principles of good design and good designers; engineering tradeoffs • Different measures for evaluation, e.g., utility, efficiency, learnability, user satisfaction	• Discuss why human-centered software development is important [Familiarity] • Define a user-centered design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances [Familiarity] • Summarize the basic precepts of psychological and social interaction [Familiarity] • Develop and use a conceptual vocabulary for analyzing human interaction with software: affordance, conceptual model, feedback, and so forth [Familiarity]
Readings : [Dix+04], [Sto+05], [RS11]	

Unit 2: Factores Humanos (8)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Cognitive models that inform interaction design, e.g., attention, perception and recognition, movement, and memory; gulfs of expectation and execution • Physical capabilities that inform interaction design, e.g., color perception, ergonomics • Accessibility, e.g., interfaces for differently-abled populations (e.g., blind, motion-impaired) • Interfaces for differently-aged population groups (e.g., children, 80+)	• Create and conduct a simple usability test for an existing software application [Familiarity]
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unit 3: User-centered design and testing (16)	
Competences Expected: b,c	
Topics	**Learning Outcomes**

• Approaches to, and characteristics of, the design process
• Functionality and usability requirements
• Techniques for gathering requirements, e.g., interviews, surveys, ethnographic and contextual enquiry
• Techniques and tools for the analysis and presentation of requirements, e.g., reports, personas
• Task analysis, including qualitative aspects of generating task analytic models
• Consideration of HCI as a design discipline
  – Sketching
  – Participatory design
  – Sketching
  – Diseño participativo
• Prototyping techniques and tools, e.g., sketching, storyboards, low-fidelity prototyping, wireframes
• Low-fidelity (paper) prototyping
• Quantitative evaluation techniques, e.g., keystroke-level evaluation
• Evaluation without users, using both qualitative and quantitative techniques, e.g., walkthroughs, GOMS, expert-based analysis, heuristics, guidelines, and standard
• Evaluation with users, e.g., observation, think-aloud, interview, survey, experiment
• Challenges to effective evaluation, e.g., sampling, generalization
• Reporting the results of evaluations
• Internationalization, designing for users from other cultures, cross-cultural
	• Conduct a quantitative evaluation and discuss/report the results [Familiarity]
	• For an identified user group, undertake and document an analysis of their needs [Familiarity]
	• Discuss at least one national or international user interface design standard [Familiarity]
	• Explain how user-centred design complements other software process models [Familiarity]
	• Use lo-fi (low fidelity) prototyping techniques to gather, and report, user responses [Usage]
	• Choose appropriate methods to support the development of a specific UI [Assessment]
	• Use a variety of techniques to evaluate a given UI [Assessment]
	• Compare the constraints and benefits of different evaluative methods [Assessment]

Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unit 4: Designing Interaction (8)	
Competences Expected: b,c,d,o	
Topics	**Learning Outcomes**
• Principles of graphical user interfaces (GUIs) • Elements of visual design (layout, color, fonts, labeling) • Handling human/system failure • User interface standards • Presenting information: navigation, representation, manipulation • Interface animation techniques (e.g., scene graphs) • Widget classes and libraries • Internationalization, designing for users from other cultures, cross-cultural • Choosing interaction styles and interaction techniques	• Create a simple application, together with help and documentation, that supports a graphical user interface [Usage]
Readings : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

<table>
<tr><td colspan="2">Unit 5: New Interactive Technologies (8)</td></tr>
<tr><td colspan="2">Competences Expected: o</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Choosing interaction styles and interaction techniques
Approaches to design, implementation and evaluation of non-mouse interaction

Touch and multi-touch interfaces
Shared, embodied, and large interfaces
New input modalities (such as sensor and location data)
New Windows, e.g., iPhone, Android
Speech recognition and natural language processing
Wearable and tangible interfaces
Persuasive interaction and emotion
Ubiquitous and context-aware interaction technologies (Ubicomp)
Bayesian inference (e.g. predictive text, guided pointing)
Ambient/peripheral display and interaction

Output

Sound
Stereoscopic display
Force feedback simulation, haptic devices

System architectures

Game engines
Mobile augmented reality
Flight simulators
CAVEs
Medical imaging

</td><td>

Describe when non-mouse interfaces are appropriate [Familiarity]
Understand the interaction possibilities beyond mouse-and-pointer interfaces [Familiarity]
Discuss the advantages (and disadvantages) of non-mouse interfaces [Usage]
Describe the optical model realized by a computer graphics system to synthesize stereoscopic view [Familiarity]
Describe the principles of different viewer tracking technologies [Familiarity]
Determine the basic requirements on interface, hardware, and software configurations of a VR system for a specified application [Assessment]

</td></tr>
<tr><td colspan="2">Readings : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]</td></tr>
</table>

Unit 6: Collaboration and communication (8)	
Competences Expected: d,o	
Topics	**Learning Outcomes**
• Asynchronous group communication, e.g., e-mail, forums, social networks • Social media, social computing, and social network analysis • Online collaboration, 'smart' spaces, and social coordination aspects of workflow technologies • Online communities • Software characters and intelligent agents, virtual worlds and avatars • Social psychology	• Describe the difference between synchronous and asynchronous communication [Familiarity] • Compare the HCI issues in individual interaction with group interaction [Familiarity] • Discuss several issues of social concern raised by collaborative software [Usage] • Discuss the HCI issues in software that embodies human intention [Assessment]
Readings : [Dix+04], [Sto+05], [RS11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.

[Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.

[Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.

[LS06] M. Leavitt and B. Shneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.

[Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.

[Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.

[RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction.* 3 ed. John Wiley and Sons Ltd, 2011.

[Sto+05] D. Stone et al. *User Interface Design and Evaluation.* Morgan Kaufmann Series in Interactive Technologies, 2005.

[WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture.* Morgan Kaufmann Publishers Inc, 2011.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS391. Software Engineering III (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS292. Software Engineering II. (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Software development requires the use of best development practices, IT project management, equipment management And efficient and rational use of quality assurance frameworks, these elements are key and transversal during the whole productive process. The construction of software contemplates the implementation and use of processes, methods, models and tools that allow to achieve the realization of the quality attributes of a product.

5. GOALS

- Understand and implement the fundamental concepts of project management and software equipment management.
- Understand the fundamentals of project management, including its definition, scope, and need for project management in the modern organization.
- Students have to understand the fundamental concepts of CMMI, PSP, TSP to be adopted in software projects.
- Describe and understand quality assurance models as a key framework for the success of IT projects.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Software Evolution (12)</td></tr>
<tr><td colspan="2">Competences Expected: c,d,i,j,m,o</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Software development in the context of large, pre-existing code bases
  – Software change
  – Concerns and concernlocation
  – Refactoring
• Software evolution
• Characteristics of maintainable software
• Reengineering systems
• Software reuse
  – Code segments
  – Libraries and frameworks
  – Components
  – Product lines
</td><td>
• Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Familiarity]
• Estimate the impact of a change request to an existing product of medium size [Usage]
• Use refactoring in the process of modifying a software component [Usage]
• Discuss the challenges of evolving systems in a changing environment [Familiarity]
• Outline the process of regression testing and its role in release management [Familiarity]
• Discuss the advantages and disadvantages of different types of software reuse [Familiarity]
</td></tr>
<tr><td colspan="2">Readings : [PM15], [Som17]</td></tr>
</table>

Unit 2: Software Project Management (10)	
Competences Expected: c,d,i,j,m,o	
Topics	**Learning Outcomes**

• Team participation
  – Team processes including responsabilities for task, meeting structure, and work schedule
  – Roles and responsabilities in a software team
  – Team conflict resolution
  – Risks associated with virtual teams (communication, perception, structure)
• Effort estimation (at the personal level)
• Risk
  – The role of risk in the lifecycle
  – Risk categories including security, safety, market, financial, technology, people, quality, structure and process
• Team management
  – Team organization and decision-making
  – Role identification and assigment
  – Individual and team performance assessment
• Project management
  – Scheduling and tracking
  – Project management tools
  – Cost/benefit analysis
	• Discuss common behaviors that contribute to the effective functioning of a team [Familiarity]
	• Create and follow an agenda for a team meeting [Usage]
	• Identify and justify necessary roles in a software development team [Usage]
	• Understand the sources, hazards, and potential benefits of team conflict [Usage]
	• Apply a conflict resolution strategy in a team setting [Usage]
	• Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage]
	• List several examples of software risks [Familiarity]
	• Describe the impact of risk in a software development lifecycle [Familiarity]
	• Describe different categories of risk in software systems [Familiarity]
	• Demonstrate through involvement in a team project the central elements of team building and team management [Usage]
	• Describe how the choice of process model affects team organizational structures and decision-making processes [Familiarity]
	• Create a team by identifying appropriate roles and assigning roles to team members [Usage]
	• Assess and provide feedback to teams and individuals on their performance in a team setting [Usage]
	• Using a particular software process, describe the aspects of a project that need to be planned and monitored, (eg, estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management) [Familiarity]

Readings : [PM15], [Som17]	

<table>
<tr><td colspan="2">Unit 3: Software Project Management (8)</td></tr>
<tr><td colspan="2">Competences Expected: c,d,i,j,m,o</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Software measurement and estimation techniques
Software quality assurance and the role of measurements
Risk

Risk identification and management
Risk analysis and evaluation
Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking)
Risk planning

System-wide approach to risk including hazards associated with tools

</td><td>

Track the progress of some stage in a project using appropriate project metrics [Usage]
Compare simple software size and cost estimation techniques [Usage]
Use a project management tool to assist in the assignment and tracking of tasks in a software development project [Usage]
Describe the impact of risk tolerance on the software development process [Assessment]
Identify risks and describe approaches to managing risk (avoidance, acceptance, transference, mitigation), and characterize the strengths and shortcomings of each [Familiarity]
Explain how risk affects decisions in the software development process [Usage]
Identify security risks for a software system [Usage]
Demonstrate a systematic approach to the task of identifying hazards and risks in a particular situation [Usage]
Apply the basic principles of risk management in a variety of simple scenarios including a security situation [Usage]
Conduct a cost/benefit analysis for a risk mitigation approach [Usage]
Identify and analyze some of the risks for an entire system that arise from aspects other than the software [Usage]

</td></tr>
<tr><td colspan="2">Readings : [PM15], [Som17]</td></tr>
</table>

Unit 4: Software Processes (12)	
Competences Expected: c,d,i,j,m,o	
Topics	**Learning Outcomes**
• System level considerations, i.e., the iteraction of software with its intended enviroment • Introduction to software process models (e.g., waterfall, incremental, agile) – Activities with software lifecycles • Programming in the large vs. individual programming • Evaluation of software process models • Software quality concepts • Process improvement • Software process capability maturity models • Software process measurements	• Describe how software can interact with and participate in various systems including information management, embedded, process control, and communications systems [Usage] • Describe the relative advantages and disadvantages among several major process models (eg, waterfall, iterative, and agile) [Usage] • Describe the different practices that are key components of various process models [Usage] • Differentiate among the phases of software development [Usage] • Describe how programming in the large differs from individual efforts with respect to understanding a large code base, code reading, understanding builds, and understanding context of changes [Usage] • Explain the concept of a software lifecycle and provide an example, illustrating its phases including the deliverables that are produced [Usage] • Compare several common process models with respect to their value for development of particular classes of software systems taking into account issues such as requirement stability, size, and non-functional characteristics [Usage] • Define software quality and describe the role of quality assurance activities in the software process [Usage] • Describe the intent and fundamental similarities among process improvement approaches [Usage] • Compare several process improvement models such as CMM, CMMI, CQI, Plan-Do-Check-Act, or ISO9000 [Usage] • Assess a development effort and recommend potential changes by participating in process improvement (using a model such as PSP) or engaging in a project retrospective [Usage] • Explain the role of process maturity models in process improvement [Usage] • Describe several process metrics for assessing and controlling a project [Usage] • Use project metrics to describe the current state of a project [Usage]
Readings : [PM15], [Som17]	

Unit 5: Estándares ISO/IEC (6)	
Competences Expected: c,d,i,j,m,o	
Topics	**Learning Outcomes**
• ISO 9001:2001. • ISO 9000-3. • ISO/IEC 9126. • ISO/IEC 12207. • ISO/IEC 15939. • ISO/IEC 14598. • ISO/IEC 15504-SPICE. • IT Mark. • SCRUM. • SQuaRE. • CISQ.	• Learn and apply correctly standards and international standards . [Usage]
Readings : [Som17], [PM15]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS401. Methodology of Computation Research (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The objective of this course is for the student to learn how to carry out scientific research in the area of computers. The teachers of the course will determine an area of study for each student, and the student will be given a bibliography to analyze. From this bibliography, and from additional bibliographic sources (researched by the student), the student should be able to construct a survey type article on the assigned topic.

5. GOALS

- That the student learns how to start a scientific investigation in the area of computing.
- That the student knows the main sources to obtain relevant bibliography for research works in the area of computing: Researchindex, IEEE-CS[1], ACM[2].
- That the student is able to analyze the existing proposals on a certain topic and relate them in a coherent way in a bibliographic review.
- That the student can write technical documents in computing using LaTeX.
- The student will be able to reproduce the existing results on a given topic through experimentation.
- The deliverables of this course are:

  Parcial advance: Mastery of the subject of the article and preliminary bibliography in article format LaTeX.

  Final: Understanding of the survey type article, concluded document containing, optionally, the experimental results of the studied technique(s).

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

[1] http://www.computer.org
[2] http://www.acm.org

Unit 1: (60)	
Competences Expected: a,b,c,i,h	
Topics	**Learning Outcomes**
• Bibliographic search in computers. • Writing technical articles on computers.	• Learn to do correct research in the area of computing. [Usage] • Knowing the sources of adequate literature for this area. [Usage] • Knowing how to write a document in accordance with the characteristics that the conferences in this area require. [Usage]
Readings : [IEE08], [Ass08], [Cit08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ass08] Association for Computing Machinery. *Digital Libray*. http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS251. Computer graphics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS312. Advanced Data Structures . (6^{th} Sem) • MA307. Mathematics applied to computing . (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It offers an introduction to the area of Computer Graphics, which is an important part of Computer Science. The purpose of this course is to investigate the fundamental principles, techniques and tools for this area.

5. GOALS

- Bring students to concepts and techniques used in complex 3-D graphics applications.
- Give the student the necessary tools to determine which graphics software and which platform are best suited to develop a specific application.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Concepts (6)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Media applications including user interfaces, audio and video editing, game engines, cad, visualization, virtual reality • Tradeoffs between storing data and re-computing data as embodied by vector and raster representations of images • Additive and subtractive color models (CMYK and RGB) and why these provide a range of colors • Animation as a sequence of still images	• Explain in general terms how analog signals can be reasonably represented by discrete samples, for example, how images can be represented by pixels [Familiarity] • Describe color models and their use in graphics display devices [Familiarity] • Describe the tradeoffs between storing information vs storing enough information to reproduce the information, as in the difference between vector and raster rendering [Familiarity] • Describe the basic process of producing continuous motion from a sequence of discrete frames (sometimes called "flicker fusion") [Familiarity]
Readings : [HB90]	

Unit 2: Basic Rendering (12)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Rendering in nature, e.g., the emission and scattering of light and its relation to numerical integration • Forward and backward rendering (i.e., ray-casting and rasterization) • Basic radiometry, similar triangles, and projection model • Affine and coordinate system transformations • Ray tracing • Visibility and occlusion, including solutions to this problem such as depth buffering, Painter's algorithm, and ray tracing • Simple triangle rasterization • Rendering with a shader-based API • Application of spatial data structures to rendering • Sampling and anti-aliasing • Forward and backward rendering (i.e., ray-casting and rasterization)	• Discuss the light transport problem and its relation to numerical integration ie, light is emitted, scatters around the scene, and is measured by the eye [Familiarity] • Describe the basic graphics pipeline and how forward and backward rendering factor in this [Familiarity] • Create a program to display 3D models of simple graphics images [Usage] • Obtain 2-dimensional and 3-dimensional points by applying affine transformations [Usage] • Apply 3-dimensional coordinate system and the changes required to extend 2D transformation operations to handle transformations in 3D [Usage] • Contrast forward and backward rendering [Assessment] • Explain the concept and applications of texture mapping, sampling, and anti-aliasing [Familiarity] • Explain the ray tracing/rasterization duality for the visibility problem [Familiarity] • Implement a simple real-time renderer using a rasterization API (eg, OpenGL) using vertex buffers and shaders [Usage] • Compute space requirements based on resolution and color coding [Assessment] • Compute time requirements based on refresh rates, rasterization techniques [Assessment]
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 3: Programming Interactive Systems (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Event management and user interaction • Approaches to design, implementation and evaluation of non-mouse interaction – Touch and multi-touch interfaces – Shared, embodied, and large interfaces – New input modalities (such as sensor and location data) – New Windows, e.g., iPhone, Android – Speech recognition and natural language processing – Wearable and tangible interfaces – Persuasive interaction and emotion – Ubiquitous and context-aware interaction technologies (Ubicomp) – Bayesian inference (e.g. predictive text, guided pointing) – Ambient/peripheral display and interaction	• Discuss the advantages (and disadvantages) of non-mouse interfaces [Assessment]
Readings : [HB90]	

<table>
<tr><td colspan="2">Unit 4: Geometric Modeling (15)</td></tr>
<tr><td colspan="2">Competences Expected: a,b,i,j</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Basic geometric operations such as intersection calculation and proximity tests
• Volumes, voxels, and point-based representations
• Parametric polynomial curves and surfaces
• Implicit representation of curves and surfaces
• Approximation techniques such as polynomial curves, Bezier curves, spline curves and surfaces, and nonuniform rational basis (NURB) spines, and level set method
• Surface representation techniques including tessellation, mesh representation, mesh fairing, and mesh generation techniques such as Delaunay triangulation, marching cubes
• Spatial subdivision techniques
• Procedural models such as fractals, generative modeling, and L-systems
• Elastically deformable and freeform deformable models
• Subdivision surfaces
• Multiresolution modeling
• Reconstruction
• Constructive Solid Geometry (CSG) representation
</td><td>
• Represent curves and surfaces using both implicit and parametric forms [Usage]
• Create simple polyhedral models by surface tessellation [Usage]
• Generate a mesh representation from an implicit surface [Usage]
• Generate a mesh from data points acquired with a laser scanner [Usage]
• Construct CSG models from simple primitives, such as cubes and quadric surfaces [Usage]
• Contrast modeling approaches with respect to space and time complexity and quality of image [Assessment]
</td></tr>
<tr><td colspan="2">Readings : [HB90], [Shr+13]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 5: Advanced Rendering (6)</td></tr>
<tr><td colspan="2">Competences Expected: a,b,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Time (motion blur), lens position (focus), and continuous frequency (color) and their impact on rendering
• Shadow mapping
• Occlusion culling
• Subsurface scattering
• Non-photorealistic rendering
• GPU architecture
• Human visual systems including adaptation to light, sensitivity to noise, and flicker fusion
</td><td>
• Demonstrate how an algorithm estimates a solution to the rendering equation [Assessment]
• Prove the properties of a rendering algorithm, eg, complete, consistent, and unbiased [Assessment]
• Implement a non-trivial shading algorithm (eg, toon shading, cascaded shadow maps) under a rasterization API [Usage]
• Discuss how a particular artistic technique might be implemented in a renderer [Familiarity]
• Explain how to recognize the graphics techniques used to create a particular image [Familiarity]
</td></tr>
<tr><td colspan="2">Readings : [HB90], [Hug+13], [Wol11], [Shr+13]</td></tr>
</table>

Unit 6: Computer Animation (4)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Forward and inverse kinematics • Collision detection and response • Procedural animation using noise, rules (boids/crowds), and particle systems • Skinning algorithms • Physics based motions including rigid body dynamics, physical particle systems, mass-spring networks for cloth and flesh and hair • Key-frame animation • Splines • Data structures for rotations, such as quaternions • Camera animation • Motion capture	• Compute the location and orientation of model parts using an forward kinematic approach [Usage] • Implement the spline interpolation method for producing in-between positions and orientations [Usage] • Implement algorithms for physical modeling of particle dynamics using simple Newtonian mechanics, for example Witkin & Kass, snakes and worms, symplectic Euler, Stormer/Verlet, or midpoint Euler methods [Usage] • Discuss the basic ideas behind some methods for fluid dynamics for modeling ballistic trajectories, for example for splashes, dust, fire, or smoke [Familiarity] • Use common animation software to construct simple organic forms using metaball and skeleton [Usage]
Readings : [HB90], [Shr+13]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C.* Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition.* Addison-Wesley, 2013.

[Shr+13] Dave Shreiner et al. *OpenGL, Programming Guide, Eighth Edition.* Addison-Wesley, 2013.

[Wol11] David Wolff. *OpenGL 4.0 Shading Language Cookbook.* Packt Publishing, 2011.

Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS262. Machine learning (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS261. Intelligent Systems. (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• ... • ...	• ... [Usage] • ... [Usage]
Readings : [De 06], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications.* CRC Press, 2006.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS2T1. Computational Biology (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS281. Computing in Society (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It offers a wide vision of the ethical and professional aspects related to computing. The topics included cover ethical, social and political aspects. The moral dimensions of computing. The methods and tools of analysis. Administration of computer resources. Security and control of computer systems. Professional and ethical responsibilities. Intellectual property.

5. GOALS

- Make the student understand the importance of care and ethics in the transfer and use of information.
- To instill in the student that the trends of technological improvement should not lead to the degradation of the morals of society.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: History (2)	
Competences Expected: f,g	
Topics	**Learning Outcomes**
• Prehistory, the world before 1946 • History of computer hardware, software, networking • Pioneers of computing • History of the Internet	• Identify significant continuing trends in the history of the computing field [Familiarity] • Identify the contributions of several pioneers in the computing field [Familiarity] • Discuss the historical context for several programming language paradigms [Familiarity] • Compare daily life before and after the advent of personal computers and the Internet [Familiarity]
Readings : [LL04], [McL00]	

Unit 2: Social Context (4)	
Competences Expected: f,g	
Topics	**Learning Outcomes**
• Social implications of computing in a networked world • Impact of social media on individualism, collectivism and culture • Growth and control of the Internet • Often referred to as the digital divide, differences in access to digital technology resources and its resulting ramifications for gender, class, ethnicity, geography, and/or underdeveloped countries • Accessibility issues, including legal requirements • Context-aware computing	• Describe positive and negative ways in which computer technology (networks, mobile computing, cloud computing) alters modes of social interaction at the personal level [Familiarity] • Identify developers' assumptions and values embedded in hardware and software design, especially as they pertain to usability for diverse populations including under-represented populations and the disabled [Usage] • Interpret the social context of a given design and its implementation [Assessment] • Evaluate the efficacy of a given design and implementation using empirical data [Familiarity] • Summarize the implications of social media on individualism versus collectivism and culture [Familiarity] • Discuss how Internet access serves as a liberating force for people living under oppressive forms of government; explain how limits on Internet access are used as tools of political and social repression [Familiarity] • Analyze the pros and cons of reliance on computing in the implementation of democracy (eg delivery of social services, electronic voting) [Familiarity] • Describe the impact of the under-representation of diverse populations in the computing profession (eg, industry culture, product diversity) [Usage] • Explain the implications of context awareness in ubiquitous computing systems [Familiarity]
Readings : [LL04], [McL00]	

Unit 3: Analytical Tools (2)	
Competences Expected: f,g,ñ	
Topics	**Learning Outcomes**
• Ethical argumentation • Ethical theories and decision-making • Moral assumptions and values	• Evaluate stakeholder positions in a given situation [Familiarity] • Analyze basic logical fallacies in an argument [Usage] • Analyze an argument to identify premises and conclusion [Familiarity] • Illustrate the use of example and analogy in ethical argument [Familiarity] • Evaluate ethical/social tradeoffs in technical decisions [Familiarity]
Readings : [LL04], [McL00]	

Unit 4: Professional Ethics (4)	
Competences Expected: f,g,ñ	
Topics	**Learning Outcomes**
• Community values and the laws by which we live • The nature of professionalism including care, attention and discipline, fiduciary responsibility, andmentoring • Keeping up-to-date as a computing professional in terms of familiarity, tools, skills, legal and professional framework as well as the ability to self-assess and progress in the computing field • Professional certification, codes of ethics, conduct, and practice, such as the ACM/IEEE-CS, SE, AITP, IFIP and international societies • Accountability, responsibility and liability (e.g. software correctness, reliability and safety, as well as ethical confidentiality of cybersecurity professionals) • The role of the computing professional in public policy • Maintaining awareness of consequences • Ethical dissent and whistle-blowing • The relationship between regional culture and ethical dilemmas • Dealing with harassment and discrimination • Forms of professional credentialing • Acceptable use policies for computing in the workplace • Ergonomics and healthy computing environments • Time to market and cost considerations versus quality professional standards	• Identify ethical issues that arise in software development and determine how to address them technically and ethically [Usage] • Explain the ethical responsibility of ensuring software correctness, reliability and safety. [Assessment] • Describe the mechanisms that typically exist for a professional to keep up-to-date [Familiarity] • Describe the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making [Familiarity] • Analyze a global computing issue, observing the role of professionals and government officials in managing this problem [Familiarity] • Evaluate the professional codes of ethics from the ACM, the IEEE Computer Society, and other organizations [Familiarity] • Describe ways in which professionals may contribute to public policy [Familiarity] • Describe the consequences of inappropriate professional behavior [Usage] • Identify progressive stages in a whistle-blowing incident [Usage] • Identify examples of how regional culture interplays with ethical dilemmas [Familiarity] • Investigate forms of harassment and discrimination and avenues of assistance [Usage] • Examine various forms of professional credentialing [Usage] • Explain the relationship between ergonomics in computing environments and people's health [Usage] • Develop a computer usage/acceptable use policy with enforcement measures [Familiarity] • Describe issues associated with industries' push to focus on time to market versus enforcing quality professional standards [Usage]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 5: Intellectual Property (4)	
Competences Expected: f,g,ñ	
Topics	**Learning Outcomes**
• Philosophical foundations of intellectual property • Intellectual property rights (cross-reference IM/Information Storage and Retrieval/intellectual property and protection) • Intangible digital intellectual property (IDIP) • Legal foundations for intellectual property protection • Digital rights management • Copyrights, patents, trade secrets, trademarks • Plagiarism • Foundations of the open source movement • Software piracy	• Discuss the philosophical bases of intellectual property [Assessment] • Discuss the rationale for the legal protection of intellectual property [Familiarity] • Describe legislation aimed at digital copyright infringements [Assessment] • Critique legislation aimed at digital copyright infringements [Familiarity] • Identify contemporary examples of intangible digital intellectual property [Assessment] • Justify uses of copyrighted materials [Assessment] [Familiarity] • Evaluate the ethical issues inherent in various plagiarism detection mechanisms [Familiarity] • Interpret the intent and implementation of software licensing [Familiarity] • Discuss the issues involved in securing software patents [Familiarity] • Characterize and contrast the concepts of copyright, patenting and trademarks [Familiarity] • Identify the goals of the open source movement [Assessment] • Identify the global nature of software piracy [Familiarity]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 6: Privacy and Civil Liberties (4)	
Competences Expected: f,g,ñ	
Topics	**Learning Outcomes**
• Philosophical foundations of privacy rights • Legal foundations of privacy protection • Privacy implications of widespread data collection for transactional databases, data warehouses, surveillance systems, and cloud computing • Ramifications of differential privacy • Technology-based solutions for privacy protection • Privacy legislation in areas of practice • Civil liberties and cultural differences • Freedom of expression and its limitations	• Discuss the philosophical basis for the legal protection of personal privacy [Familiarity] • Evaluate solutions to privacy threats in transactional databases and data warehouses [Familiarity] • Describe the role of data collection in the implementation of pervasive surveillance systems (e.g., RFID, face recognition, toll collection, mobile computing). [Familiarity] • Describe the ramifications of differential privacy. [Familiarity] • Investigate the impact of technological solutions to privacy problems [Familiarity] • Critique the intent, potential value and implementation of various forms of privacy legislation [Familiarity] • Identify strategies to enable appropriate freedom of expression [Familiarity]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 7: Security Policies, Laws and Computer Crimes (2)	
Competences Expected: f,g,ñ	
Topics	**Learning Outcomes**
• Examples of computer crimes and legal redress for computer criminals • Social engineering, identity theft and recovery • Issues surrounding the misuse of access and breaches in security • Motivations and ramifications of cyber terrorism and criminal hacking, "cracking" • Effects of malware, such as viruses, worms and Trojan horses • Crime prevention strategies • Security policies	• List classic examples of computer crimes and social engineering incidents with societal impact [Familiarity] • Identify laws that apply to computer crimes [Familiarity] • Describe the motivation and ramifications of cyber terrorism and criminal hacking [Familiarity] • Examine the ethical and legal issues surrounding the misuse of access and various breaches in security [Familiarity] • Discuss the professional's role in security and the trade-offs involved [Familiarity] • Investigate measures that can be taken by both individuals and organizations including governments to prevent or mitigate the undesirable effects of computer crimes and identity theft [Familiarity] • Write a company-wide security policy, which includes procedures for managing passwords and employee monitoring [Familiarity]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 8: Economies of Computing (2)	
Competences Expected: f,g,ñ,o	
Topics	**Learning Outcomes**
• Monopolies and their economic implications • Effect of skilled labor supply and demand on the quality of computing products • Pricing strategies in the computing domain • The phenomenon of outsourcing and off-shoring software development; impacts on employment and on economics • Consequences of globalization for the computer science profession • Differences in access to computing resources and the possible effects thereof • Cost/benefit analysis of jobs with considerations to manufacturing, hardware, software, and engineering implications • Cost estimates versus actual costs in relation to total costs • Entrepreneurship: prospects and pitfalls • Network effect or demand-side economies of scale • Use of engineering economics in dealing with finances	• Summarize the rationale for antimonopoly efforts [Familiarity] • Identify several ways in which the information technology industry is affected by shortages in the labor supply [Familiarity] • Identify the evolution of pricing strategies for computing goods and services [Familiarity] • Discuss the benefits, the drawbacks and the implications of off-shoring and outsourcing [Familiarity] • Investigate and defend ways to address limitations on access to computing [Usage] • Describe the economic benefits of network effects [Usage]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Edi09a] Datamation Ediciones, ed. *Revista Datamation MC Ediciones*. 2009.

[Edi09b] Datamation Ediciones, ed. *Understanding the Digital Economy*. 2009.

[Edi10] Datamation Ediciones, ed. *Financial Times Mastering Information Management*. 2010.

[LL04] Kenneth C. Laudon and Jane P. Laudon. *Sistemas de Información Gerencial*. Prentice Hall, 2004.

[McL00] Raymond McLeod Jr. *Sistemas de Información Gerencial*. Prentice Hall, 2000.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3I1. Computer Security (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS231. Networking and Communication. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Nowadays, information is one of the most valuable assets in any organization. This course is oriented to be able to provide the student with the security elements oriented to protect the Information of the organization and mainly to be able to foresee the possible problems related to this heading. This subject involves the development of a preventive attitude on the part of the student in all areas related to software development.

5. GOALS

- Discuss at an intermediate intermediate level the fundamentals of Computer Security.
- Provide different aspects of the malicious code.
- That the student knows the concepts of cryptography and security in computer networks.
- Discuss and analyze together with the student the aspects of Internet Security.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Foundational Concepts in Security (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**
• CIA (Confidentiality, Integrity, Availability) • Concepts of risk, threats, vulnerabilities, and attack vectors • Authentication and authorization, access control (mandatory vs. discretionary) • Concept of trust and trustworthiness • Ethics (responsible disclosure)	• Analyze the tradeoffs of balancing key security properties (Confidentiality, Integrity, Availability) [Familiarity] • Describe the concepts of risk, threats, vulnerabilities and attack vectors (including the fact that there is no such thing as perfect security) [Familiarity] • Explain the concepts of authentication, authorization, access control [Familiarity] • Explain the concept of trust and trustworthiness [Familiarity] • Recognize that there are important ethical issues to consider in computer security, including ethical issues associated with fixing or not fixing vulnerabilities and disclosing or not disclosing vulnerabilities [Familiarity]
Readings : [WL14]	

Unit 2: Principles of Secure Design (25)	
Competences Expected: g,a,e,h	
Topics	**Learning Outcomes**
• Least privilege and isolation • Fail-safe defaults • Open design • End-to-end security • Defense in depth (e.g., defensive programming, layered defense) • Security by design • Tensions between security and other design goals • Complete mediation • Use of vetted security components • Economy of mechanism (reducing trusted computing base, minimize attack surface) • Usable security • Security composability • Prevention, detection, and deterrence	• Describe the principle of least privilege and isolation as applied to system design [Familiarity] • Summarize the principle of fail-safe and deny-by-default [Familiarity] • Discuss the implications of relying on open design or the secrecy of design for security. [Familiarity] • Explain the goals of end-to-end data security [Familiarity] • Discuss the benefits of having multiple layers of defenses [Familiarity] • For each stage in the lifecycle of a product, describe what security considerations should be evaluated. [Familiarity] • Describe the cost and tradeoffs associated with designing security into a product [Familiarity] • Describe the concept of mediation and the principle of complete mediation [Familiarity] • Be aware of standard components for security operations, instead of re-inventing fundamentals operations [Familiarity] • Explain the concept of trusted computing including trusted computing base and attack surface and the principle of minimizing trusted computing base [Familiarity] • Discuss the importance of usability in security mechanism design [Familiarity] • Describe security issues that arise at boundaries between multiple components. [Familiarity] • Identify the different roles of prevention mechanisms and detection/deterrence mechanisms [Familiarity]
Readings : [WL14]	

Unit 3: Defensive Programming (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**

• Input validation and data sanitization
• Choice of programming language and type-safe languages
• Examples of input validation and data sanitization errors
  – Buffer overflows
  – Integer errors
  – SQL injection
  – XSS vulnerability
• Race conditions
• Correct handling of exceptions and unexpected behaviors
• Correct usage of third-party components
• Effectively deploying security updates
• Information flow control
• Correctly generating randomness for security purposes
• Mechanisms for detecting and mitigating input and data sanitization errors
• Fuzzing
• Static analysis and dynamic analysis
• Program verification
• Operating system support (e.g., address space randomization, canaries)
• Hardware support (e.g, DEP, TPM)
	• Explain why input validation and data sanitization is necessary in the face of adversarial control of the input channel. [Usage]
	• Explain why you might choose to develop a program in a type-safe language like Java, in contrast to an unsafe programming language like C/C++ [Usage]
	• Classify common input validation errors, and write correct input validation code [Usage]
	• Demonstrate using a high-level programming language how to prevent a race condition from occurring and how to handle an exception [Usage]
	• Demonstrate the identification and graceful handling of error conditions [Familiarity]
	• Explain the risks with misusing interfaces with third-party code and how to correctly use third-party code [Familiarity]
	• Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix [Familiarity]

Readings : [WL14]	

Unit 4: Threats and Attacks (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Attacker goals, capabilities, and motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats) • Examples of malware (e.g., viruses, worms, spyware, botnets, Trojan horses or rootkits) • Denial of Service (DoS) and Distributed Denial of Service (DDoS) • Social engineering (e.g., phishing) • Attacks on privacy and anonymity • Malware/unwanted communication such as covert channels and steganography	• Describe likely attacker types against a particular system [Familiarity] • Discuss the limitations of malware countermeasures (eg, signature-based detection, behavioral detection) [Familiarity] • Identify instances of social engineering attacks and Denial of Service attacks [Familiarity] • Discuss how Denial of Service attacks can be identified and mitigated [Familiarity] • Describe risks to privacy and anonymity in commonly used applications [Familiarity] • Discuss the concepts of covert channels and other data leakage procedures [Familiarity]
Readings : [WL14]	

Unit 5: Network Security (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, man-in-the-middle, message integrity attacks, routing attacks, and traffic analysis) • Use of cryptography for data and network security • Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, anonymous communication protocols, isolation) • Defense mechanisms and countermeasures (e.g., network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks) • Security for wireless, cellular networks • Other non-wired networks (e.g., ad hoc, sensor, and vehicular networks) • Censorship resistance • Operational network security management (e.g., configure network access control)	• Describe the different categories of network threats and attacks [Familiarity] • Describe the architecture for public and private key cryptography and how PKI supports network security [Familiarity] • Describe virtues and limitations of security technologies at each layer of the network stack [Familiarity] • Identify the appropriate defense mechanism(s) and its limitations given a network threat [Usage]
Readings : [WL14]	

<table>
<tr><td colspan="2">Unit 6: Cryptography (25)</td></tr>
<tr><td colspan="2">Competences Expected: b,e,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Basic Cryptography Terminology covering notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures
Cipher types (e.g., Caesar cipher, affine cipher) together with typical attack methods such as frequency analysis
Public Key Infrastructure support for digital signature and encryption and its challenges
Symmetric key cryptography

Perfect secrecy and the one time pad
Modes of operation for semantic security and authenticated encryption (e.g., encrypt-then-MAC, OCB, GCM)
Message integrity (e.g., CMAC, HMAC)

Public key cryptography:

Trapdoor permutation, e.g., RSA
Public key encryption, e.g., RSA encryption, EI Gamal encryption
Digital signatures
Public-key infrastructure (PKI) and certificates
Hardness assumptions, e.g., Diffie-Hellman, integer factoring

Authenticated key exchange protocols, e.g., TLS
Cryptographic primitives:

pseudo-random generators and stream ciphers
block ciphers (pseudo-random permutations), e.g., AES
pseudo-random functions
hash functions, e.g., SHA2, collision resistance
message authentication codes
key derivations functions

</td><td>

Describe the purpose of Cryptography and list ways it is used in data communications [Familiarity]
Define the following terms: Cipher, Cryptanalysis, Cryptographic Algorithm, and Cryptology and describe the two basic methods (ciphers) for transforming plain text in cipher text [Familiarity]
Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Familiarity]
Illustrate how to measure entropy and how to generate cryptographic randomness [Usage]
Use public-key primitives and their applications [Usage]
Explain how key exchange protocols work and how they fail [Familiarity]
Discuss cryptographic protocols and their properties [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [WL14]</td></tr>
</table>

Unit 7: Web Security (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**

• Web security model
  – Browser security model including same-origin policy
  – Client-server trust boundaries, e.g., cannot rely on secure execution in the client
• Session management, authentication
  – Single sign-on
  – HTTPS and certificates
• Application vulnerabilities and defenses
  – SQL injection
  – XSS
  – CSRF
• Client-side security
  – Cookies security policy
  – HTTP security extensions, e.g. HSTS
  – Plugins, extensions, and web apps
  – Web user tracking
• Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers
	• Describe the browser security model including same-origin policy and threat models in web security [Familiarity]
	• Discuss the concept of web sessions, secure communication channels such as TLS and importance of secure certificates, authentication including single sign-on such as OAuth and SAML [Familiarity]
	• Investigate common types of vulnerabilities and attacks in web applications, and defenses against them [Familiarity]
	• Use client-side security capabilities [Usage]

Readings : [WL14]	

<table>
<tr><td colspan="2">Unit 8: Platform Security (25)</td></tr>
<tr><td colspan="2">Competences Expected: b,e,i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Code integrity and code signing
Secure boot, measured boot, and root of trust
Attestation
TPM and secure co-processors
Security threats from peripherals, e.g., DMA, IOMMU
Physical attacks: hardware Trojans, memory probes, cold boot attacks
Security of embedded devices, e.g., medical devices, cars
Trusted path

</td><td>

Explain the concept of code integrity and code signing and the scope it applies to [Familiarity]
Discuss the concept of root of trust and the process of secure boot and secure loading [Familiarity]
Describe the mechanism of remote attestation of system integrity [Familiarity]
Summarize the goals and key primitives of TPM [Familiarity]
Identify the threats of plugging peripherals into a device [Familiarity]
Identify physical attacks and countermeasures [Familiarity]
Identify attacks on non-PC hardware platforms [Familiarity]
Discuss the concept and importance of trusted path [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [WL14]</td></tr>
</table>

Unit 9: Digital Forensics (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**
• Basic Principles and methodologies for digital forensics • Design systems with forensic needs in mind • Rules of Evidence - general concepts and differences between jurisdictions and Chain of Custody • Search and Seizure of evidence: legal and procedural requirements • Digital Evidence methods and standards • Techniques and standards for Preservation of Data • Legal and Reporting Issues including working as an expert witness • OS/File System Forensics • Application Forensics • Web Forensics • Network Forensics • Mobile Device Forensics • Computer/network/system attacks • Attack detection and investigation • Anti-forensics	• Describe what is a Digital Investigation is, the sources of digital evidence, and the limitations of forensics [Familiarity] • Explain how to design software to support forensics [Familiarity] • Describe the legal requirements for use of seized data [Familiarity] • Describe the process of evidence seizure from the time when the requirement was identified to the disposition of the data [Familiarity] • Describe how data collection is accomplished and the proper storage of the original and forensics copy [Familiarity] • Conduct data collection on a hard drive [Usage] • Describe a person's responsibility and liability while testifying as a forensics examiner [Familiarity] • Recover data based on a given search term from an imaged system [Usage] • Reconstruct application history from application artifacts [Familiarity] • Reconstruct web browsing history from web artifacts [Familiarity] • Capture and interpret network traffic [Familiarity] • Discuss the challenges associated with mobile device forensics [Familiarity]
Readings : [WL14]	

Unit 10: Secure Software Engineering (25)	
Competences Expected: a,g,i,c	
Topics	**Learning Outcomes**
• Building security into the software development lifecycle • Secure design principles and patterns • Secure software specifications and requirements • Secure software development practices • Secure testing- the process of testing that security requirements are met (including static and dynamic analysis).	• Describe the requirements for integrating security into the SDL [Familiarity] • Apply the concepts of the Design Principles for Protection Mechanisms, the Principles for Software Security (Viega and McGraw), and the Principles for Secure Design (Morrie Gasser) on a software development project [Familiarity] • Develop specifications for a software development effort that fully specify functional requirements and identifies the expected execution paths [Familiarity]
Readings : [WL14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3P1. Parallel and Distributed Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS212. Analysis and Design of Algorithms. (5^{th} Sem) • CS231. Networking and Communication. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The last decade has brought explosive growth in computing with multiprocessors, including Multi-core processors and distributed data centers. As a result, computing parallel and distributed has become a widely elective subject to be one of the main components in the mesh studies in computer science undergraduate. Both parallel and distributed computing the simultaneous execution of multiple processes, whose operations have the potential to intercalar in a complex way. Parallel and distributed computing builds on foundations in many areas, including understanding the fundamental concepts of systems, such as: concurrency and parallel execution, consistency in state / memory manipulation, and latency. The communication and coordination between processes has its foundations in the passage of messages and models of shared memory of computing and algorithmic concepts like atomicity, consensus and conditional waiting.Achieving acceleration in practice requires an understanding of parallel algorithms, strategies for decomposition problem, systems architecture, implementation strategies and analysis of performance. Distributed systems highlight the problems of security and tolerance to Failures, emphasize the maintenance of the replicated state and introduce additional problems in the field of computer networks.

5. GOALS

- That the student is able to create parallel applications of medium complexity by efficiently leveraging machines with multiple cores.
- That the student is able to compare sequential and parallel applications.
- That the student is able to convert, when the situation warrants, sequential applications to parallel efficiently

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><th colspan="2">Unit 1: Parallelism Fundamentals (18)</th></tr>
<tr><th colspan="2">Competences Expected: a</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

Multiple simultaneous computations
Goals of parallelism (e.g., throughput) versus concurrency (e.g., controlling access to shared resources)
Parallelism, communication, and coordination

Parallelism, communication, and coordination
Need for synchronization

Programming errors not found in sequential programming

Data races (simultaneous read/write or write/write of shared state)
Higher-level races (interleavings violating program intention, undesired non-determinism)
Lack of liveness/progress (deadlock, starvation)

</td><td>

Distinguish using computational resources for a faster answer from managing efficient access to a shared resource [Familiarity]
Distinguish multiple sufficient programming constructs for synchronization that may be inter-implementable but have complementary advantages [Familiarity]
Distinguish data races from higher level races [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Pac11], [Mat14], [quinnz], [Geo10]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 2: Parallel Architecture (12)</th></tr>
<tr><th colspan="2">Competences Expected: a,b</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>

Multicore processors
Shared vs distributed memory
Symmetric multiprocessing (SMP)
SIMD, vector processing
GPU, co-processing
Flynn's taxonomy
Instruction level support for parallel programming

Atomic instructions such as Compare and Set

Memory issues

Multiprocessor caches and cache coherence
Non-uniform memory access (NUMA)

Topologies

Interconnects
Clusters
Resource sharing (e.g., buses and interconnects)

</td><td>

Explain the differences between shared and distributed memory [Assessment]
Describe the SMP architecture and note its key features [Assessment]
Characterize the kinds of tasks that are a natural match for SIMD machines [Usage]
Describe the advantages and limitations of GPUs vs CPUs [Usage]
Explain the features of each classification in Flynn's taxonomy [Usage]
Describe the challenges in maintaining cache coherence [Familiarity]
Describe the key performance challenges in different memory and distributed system topologies [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Pac11], [KH13], [SK10], [Geo10]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 3: Parallel Decomposition (18)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Need for communication and coordination/synchronization
• Independence and partitioning
• Basic knowledge of parallel decomposition concept
• Task-based decomposition
  – Implementation strategies such as threads
• Data-parallel decomposition
  – Strategies such as SIMD and MapReduce
• Actors and reactive processes (e.g., request handlers)
</td><td>
• Explain why synchronization is necessary in a specific parallel program [Usage]
• Identify opportunities to partition a serial program into independent parallel modules [Familiarity]
• Write a correct and scalable parallel algorithm [Usage]
• Parallelize an algorithm by applying task-based decomposition [Usage]
• Parallelize an algorithm by applying data-parallel decomposition [Usage]
• Write a program using actors and/or reactive processes [Usage]
</td></tr>
<tr><td colspan="2">Readings : [Pac11], [Mat14], [Qui03], [Geo10]</td></tr>
</table>

Unit 4: Communication and Coordination (18)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Shared Memory
• Consistency, and its role in programming language guarantees for data-race-free programs
• Message passing
  – Point-to-point versus multicast (or event-based) messages
  – Blocking versus non-blocking styles for sending and receiving messages
  – Message buffering (cross-reference PF/Fundamental Data Structures/Queues)
• Atomicity
  – Specifying and testing atomicity and safety requirements
  – Granularity of atomic accesses and updates, and the use of constructs such as critical sections or transactions to describe them
  – Mutual Exclusion using locks, semaphores, monitors, or related constructs
    ∗ Potential for liveness failures and deadlock (causes, conditions, prevention)
  – Composition
    ∗ Composing larger granularity atomic actions using synchronization
    ∗ Transactions, including optimistic and conservative approaches
• Consensus
  – (Cyclic) barriers, counters, or related constructs
• Conditional actions
  – Conditional waiting (e.g., using condition variables)
	• Use mutual exclusion to avoid a given race condition [Usage]
	• Give an example of an ordering of accesses among concurrent activities (eg, program with a data race) that is not sequentially consistent [Familiarity]
	• Give an example of a scenario in which blocking message sends can deadlock [Usage]
	• Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity]
	• Write a program that correctly terminates when all of a set of concurrent tasks have completed [Usage]
	• Give an example of a scenario in which an attempted optimistic update may never complete [Familiarity]
	• Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]

Readings : [Pac11], [Mat14], [Qui03], [Geo10]	

Unit 5: Parallel Algorithms, Analysis, and Programming (18)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Critical paths, work and span, and the relation to Amdahl's law • Speed-up and scalability • Naturally (embarrassingly) parallel algorithms • Parallel algorithmic patterns (divide-and-conquer, map and reduce, master-workers, others) – Specific algorithms (e.g., parallel MergeSort) • Parallel graph algorithms (e.g., parallel shortest path, parallel spanning tree) (cross-reference AL/Algorithmic Strategies/Divide-and-conquer) • Parallel matrix computations • Producer-consumer and pipelined algorithms • Examples of non-scalable parallel algorithms	• Define "critical path", "work", and "span" [Familiarity] • Compute the work and span, and determine the critical path with respect to a parallel execution diagram [Usage] • Define "speed-up" and explain the notion of an algorithm's scalability in this regard [Familiarity] • Identify independent tasks in a program that may be parallelized [Usage] • Characterize features of a workload that allow or prevent it from being naturally parallelized [Familiarity] • Implement a parallel divide-and-conquer (and/or graph algorithm) and empirically measure its performance relative to its sequential analog [Usage] • Decompose a problem (eg, counting the number of occurrences of some word in a document) via map and reduce operations [Usage] • Provide an example of a problem that fits the producer-consumer paradigm [Usage] • Give examples of problems where pipelining would be an effective means of parallelization [Usage] • Implement a parallel matrix algorithm [Usage] • Identify issues that arise in producer-consumer algorithms and mechanisms that may be used for addressing them [Usage]
Readings : [Mat14], [Qui03], [Geo10]	

Unit 6: Parallel Performance (18)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**

• Load balancing
• Performance measurement
• Scheduling and contention (cross-reference OS/Scheduling and Dispatch)
• Evaluating communication overhead
• Data management
  – Non-uniform communication costs due to proximity (cross-reference SF/Proximity)
  – Cache effects (e.g., false sharing)
  – Maintaining spatial locality
• Power usage and management
	• Detect and correct a load imbalance [Usage]
	• Calculate the implications of Amdahl's law for a particular parallel algorithm (cross-reference SF/Evaluation for Amdahl's Law) [Usage]
	• Describe how data distribution/layout can affect an algorithm's communication costs [Familiarity]
	• Detect and correct an instance of false sharing [Usage]
	• Explain the impact of scheduling on parallel performance [Familiarity]
	• Explain performance impacts of data locality [Familiarity]
	• Explain the impact and trade-off related to power usage on parallel performance [Familiarity]

Readings : [Pac11], [Mat14], [KH13], [SK10], [Geo10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Geo10] Gerhard Wellein Georg Hager. *Introduction to High Performance Computing for Scientists and Engineers (Chapman & HallCRC Computational Science)*. Ed. by CRC Press. 1st. 2010. ISBN: 978-1439811924.

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

[Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: `http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf`.

[Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.

[Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS402. Capstone Project I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS401. Methodology of Computation Research . (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to allow the student to carry out a study of the state of the art of a topic chosen by the student for his thesis.

5. GOALS

- That the student carries out an initial investigation in a specific subject realizing the study of the state of the art of the chosen subject.
- That the student shows mastery in the subject of the line of investigation chosen
- That the student choose a teacher who dominates the research chosen as an advisor.
- The deliverables of this course are:

  Avance parcial: Solid bibliography and progress of a Technical Reporto.

  Final: Technical Report with preliminary comparative experiments that demonstrate that the student already knows the existing techniques in the area of his project and choose a teacher who dominates the area of his project as an adviser of his project.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Lifting the state of the art (60)	
Competences Expected: e,h,i,l	
Topics	**Learning Outcomes**
• Perform an in-depth study of the state of the art in a certain topic in the area of Computation. • Writing technical articles in computing.	• Make a bibliographical survey of the state of the art of the chosen subject (this probably means 1 or 2 chapters of theoretical framework in addition to the introduction that is chapter I of the thesis) [Usage] • Writing a latex document in paper format with higher quality than Project I (master tables, figures, equations, indices, bibtex, cross references, citations, pstricks) [Usage] • Try to make presentations using prosper [Usage] • Show basic experiments [Usage] • Choose an advisor who dominates the research area [Usage]
Readings : [IEE08], [Ass08], [Cit08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ass08] Association for Computing Machinery. *Digital Libray*. http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS361. Computational Vision (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• ... • ...	• ... [Usage] • ... [Usage]
Readings : [De 06], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications.* CRC Press, 2006.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS371. Data Analysis (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2702. Data Management II. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Usage] • ... [Usage] • ... [Usage]
Readings : [Bur04], [Cel05]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T1. Information Processing in Biological Cells (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T2. Omic Data Modeling (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

ET201. Entrerpreneurship I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG350. Leadership and Performance. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the first course in the area of training for technological basis, aims to provide the future professional of knowledge, attitudes and skills that will allow a business plan to be drawn up for a technology-based company. The course is divided into the following units: Introduction, Creativity, From Idea to Opportunity, The Canvas Model, Customer Development and Lean Startup, Legal Aspects and Marketing, Company Finance and Presentation.

The aim is to take advantage of the creative and innovative potential and effort of the students in the creation of new companies.

5. GOALS

- That the student knows how to prepare a business plan to start a technology-based company.
- That the student is able to carry out, using business models, the conception and presentation of a business proposal.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (5)	
Competences Expected: C2	
Topics	**Learning Outcomes**
• Entrepreneurship, entrepreneurship and technological innovation. • Business models. • Team building.	• Identify characteristics of entrepreneurs. [Familiarity] • Introducing business models. [Familiarity]
Readings : [BDN10], [OP10], [Gar+14]	

Unit 2: (5)	
Competences Expected: C10	
Topics	**Learning Outcomes**
• Vision. • Mission. • The Value Proposition. • Creativity and invention. • Types and sources of innovation. • Strategy and Technology. • Scale and scope.	• Correctly setting out the company's vision and mission. [Usage] • Characterize an innovative value proposition. [Assessment] • Identify the various types and sources of innovation. [Familiarity]
Readings : [BDN10], [BD12], [Gar+14]	

Unit 3: (5)	
Competences Expected: C17	
Topics	**Learning Outcomes**
• Company Strategy. • Barriers . • Sustainable competitive advantage. • Alliances. • Organizational learning. • Product development and design.	• Knowing business strategies. [Familiarity] • Characterize barriers and competitive advantages. [Familiarity]
Readings : [BDN10], [OP10], [Rie11], [Gar+14]	

Unit 4: (20)	
Competences Expected: C18	
Topics	**Learning Outcomes**
• Creating a new business. • The business plan. • Canvas. • Elements of the Canvas.	• Get to know the elements of the Canvas model. [Usage] • Develop a business plan based on the Canvas model. [Usage]
Readings : [OP10], [BD12], [Gar+14]	

Unit 5: (20)	
Competences Expected: C19	
Topics	**Learning Outcomes**
• Acceleration versus incubation. • Customer Development. • Lean Startup.	• Knowing and applying the Customer Development model. [Usage] • Knowing and applying the Lean Startup model. [Usage]
Readings : [BD12], [Rie11], [Gar+14]	

Unit 6: (5)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Legal and tax aspects for the incorporation of the company. • Intellectual Property. • Patents. • Copyrights and trademarks. • Marketing objectives and market segments. • Market research and customer search.	• Knowing the legal aspects necessary for the formation of a technology company. [Familiarity] • Identify market segments and marketing objectives. [Familiarity]
Readings : [BDN10], [Rie11], [Con96], [Rep97], [Gar+14]	

Unit 7: (5)	
Competences Expected: C23	
Topics	**Learning Outcomes**
• Cost model. • Utility Model. • Price. • Financial Plan. • Ways of financing. • Sources of capital. • Venture Capital.	• Define a cost and profit model. [Assessment] • Knowing the various sources of funding. [Familiarity]
Readings : [BDN10], [BD12], [Gar+14]	

Unit 8: (5)	
Competences Expected: CS5	
Topics	**Learning Outcomes**
• The Elevator Pitch. • Presentation. • Negotiation.	• Knowing the different ways to present business proposals. [Familiarity] • Make the presentation of a business proposal. [Usage]
Readings : [BDN10], [BD12], [Gar+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[BD12] Steve Blank and Bob Dorf. *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company.* K and S Ranch, 2012.

[BDN10] Thomas Byers, Richard Dorf, and Andrew Nelson. *Technology Ventures: From Idea to Enterprise.* McGraw-Hill Science, 2010.

[Con96] Congreso de la Republica del Perú. *Decreto Legislativo Nº823. Ley de la Propiedad Industrial.* El Peruano, 1996.

[Gar+14] René Garzozi-Pincay et al. *Planes de Negocios para Emprendedores.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

[OP10] Alexander Osterwalder and Yves Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers.* Wiley, 2010.

[Rep97] Congreso de la Republica del Peru. *Ley Nº26887. Ley General de Sociedades.* El Peruano, 1997.

[Rie11] Eric Ries. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* Crown Business, 2011.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS370. Big Data (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS2702. Data Management II. (5^{th} Sem) • CS3P1. Parallel and Distributed Computing . (8^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Nowadays, knowing scalable approaches to processing and storing large volumes of information (terabytes, petabytes and even exabytes) is fundamental in computer science courses. Every day, every hour, every minute generates a large amount of information which needs to be processed, stored, analyzed.

5. GOALS

- That the student is able to create parallel applications to process large volumes of information
- That the student is able to compare the alternatives for the processing of big data
- That the student is able to propose architectures for a scalable application

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introducción a Big Data (15)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Overview on Cloud Computing • Distributed File System Overview • Overview of the MapReduce programming model	• Explain the concept of Cloud Computing from the point of view of Big Data[Familiarity] • Explain the concept of Distributed File System [Familiarity] • Explain the concept of the MapReduce programming model[Familiarity]
Readings : [Cou+11]	

Unit 2: Hadoop (15)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Hadoop overview. • History. • Hadoop Structure. • HDFS, Hadoop Distributed File System. • Programming Model MapReduce	• Understand and explain the Hadoop suite [Familiarity] • Implement solutions using the MapReduce programming model. [Usage] • Understand how data is saved in the HDFS. [Familiarity]
Readings : [HDF11], [BVS13]	

Unit 3: Procesamiento de Grafos en larga escala (10)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Pregel: A System for Large-scale Graph Processing. • Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. • Apache Giraph is an iterative graph processing system built for high scalability.	• Understand and explain the architecture of the Pregel project. [Familiarity] • Understand the GraphLab project architecture. [Familiarity] • Understand the architecture of the Giraph project. [Familiarity] • Implement solutions using Pregel, GraphLab or Giraph. [Usage]
Readings : [Low+12], [Mal+10], [Bal+08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: http://doi.acm.org/10.1145/1367497.1367618.

[BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.

[Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.

[HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.

[Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: http://dx.doi.org/10.14778/2212351.2212354.

[Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *ACM SIGMOD Record.* SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: http://doi.acm.org/10.1145/1807167.1807184.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS403. Final Project II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS402. Capstone Project I. (8^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims at the student to conclude his thesis project.

5. GOALS

- That the student is in the capacity to formally present his thesis project with the theoretical framework and complete bibliographic survey.
- That the student master the state of the art of his area of research.
- The deliverables of this course are:

Avance parcial: Thesis plan progress including motivation and context, problem definition, objectives, schedule of activities up to the final thesis project and the state of the art of the topic addressed.

Final: Complete thesis plan and advancement of Thesis including theoretical framework chapters, related works and preliminary (formal or statistical) results oriented to your thesis topic.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Thesis project (30)	
Competences Expected: a,b,c,e,f,h,i,l	
Topics	**Learning Outcomes**
• Thesis project.	• Description of the format used by the University for the thesis[Assessment] • Conclude the thesis project plan[Assessment] • Present the state of the art thesis topic(50%)[Assessment]
Readings : [IEE08], [Ass08], [Cit08]	

Unit 2: Thesis progress (30)	
Competences Expected: a,b,c,e,f,h,i,l	
Topics	**Learning Outcomes**
• Thesis Progress.	• Description of the format used by the University for the thesis[Assessment] • Conclude the chapter of the theoretical framework of the Thesis[Assessment] • Complete the chapter on related works(35%)[Assessment] • Plan, develop and present results (formal or statistical) of experiments oriented to your thesis topic (35%)[Assessment]
Readings : [IEE08], [Ass08], [Cit08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ass08] Association for Computing Machinery. *Digital Libray*. http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS351. Topics in Computer Graphics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS251. Computer graphics . (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In this course you can delve into any of the topics Mentioned in the area of Graphics Computing (Graphics and Visual Computing - GV).

This course is designed to perform some advanced course suggested by the ACM / IEEE curriculum. [Hug+13; HB90]

5. GOALS

- That the student uses computer techniques Graphs that involve complex data structures and algorithms.
- That the student apply the concepts learned to create an application about a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Advanced Topics on Computer Graphics (0)	
Competences Expected: a,b,m	
Topics	**Learning Outcomes**
• CS355. Advanced Computer Graphics • CS356. Computer animation • CS313. Geometric Algorithms • CS357. visualization • CS358. Virtual reality • CS359. Genetic algorithms	• Advanced Topics on Computer Graphics
Readings : [**Soars022S**], [**Soars022W**], [**Soars022T**], [**Cambridge06**], [**MacGrew99**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS362. Natural Language Processing (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• ... • ...	• ... [Usage] • ... [Usage]
Readings : [De 06], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS363. Learning by Reinforcement (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• ... • ...	• ... [Usage] • ... [Usage]
Readings : [De 06], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS364. Cognitive Computing (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• ... • ...	• ... [Usage] • ... [Usage]
Readings : [De 06], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications.* CRC Press, 2006.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS372. Web mining (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2702. Data Management II. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Usage] • ... [Usage] • ... [Usage]
Readings : [Bur04], [Cel05]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS373. Data Visualization (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2702. Data Management II. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Usage] • ... [Usage] • ... [Usage]
Readings : [Bur04], [Cel05]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS392. Tópicos en Ingeniería de Software (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS391. Software Engineering III. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Software development requires the use of best development practices, IT project management, team management and efficient and rational use of quality assurance and portfolio management frameworks, these elements are part key and transversal for the success of the production process.

This course explores the design, selection, implementation and management of IT solutions in Organizations. The focus is on applications and infrastructure and their application in the business.

5. GOALS

- Understand a variety of frameworks for enterprise architecture analysis and decision making.
- Use techniques to evaluate and manage risk in the company's portfolio.
- Assess and plan the integration of emerging technologies.
- Understand the role and potential of IT to support business process management.
- Understand the different approaches to modeling and improving business processes.
- Describe and understand quality assurance models as a key framework for successful IT projects.
- Understand and apply the IT Governance framework as a key element in managing the Enterprise application portfolio.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Software Design (18)	
Competences Expected: c,d,i,j,m,o	
Topics	**Learning Outcomes**

• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures
• Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented
• Structural and behavioral models of software designs
• Design patterns
• Relationships between requirements and designs: transformation of models, design of contracts, invariants
• Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters)
• The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set)
• Refactoring designs using design patterns
• Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments
• Measurement and analysis of design quality
• Tradeoffs between different aspects of quality
• Application frameworks
• Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems
• Principles of secure design and coding
  – Principle of least privilege
  – Principle of fail-safe defaults
  – Principle of psychological acceptability
	• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Usage]
	• Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage]
	• Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage]
	• Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Usage]
	• For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage]
	• Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage]
	• Explain the relationships between the requirements for a software product and its design, using appropriate models [Usage]
	• For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Usage]
	• Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Usage]
	• Investigate the impact of software architectures selection on the design of a simple system [Usage]
	• Apply simple examples of patterns in a software design [Usage]
	• Describe a form of refactoring and discuss when it may be applicable [Usage]
	• Select suitable components for use in the design of a software product [Usage]
	• Explain how suitable components might need to be adapted for use in the design of a software product [Usage]
	• Design a contract for a typical small software component for use in a given system [Usage]
	• Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage]
	• Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-

Unit 2: Software Project Management (14)

Competences Expected: c,d,i,j,m,o

Topics

- Team participation
  - Team processes including responsabilities for task, meeting structure, and work schedule
  - Roles and responsabilities in a software team
  - Team conflict resolution
  - Risks associated with virtual teams (communication, perception, structure)
- Effort estimation (at the personal level)
- Risk
  - The role of risk in the lifecycle
  - Risk categories including security, safety, market, financial, technology, people, quality, structure and process
- Team management
  - Team organization and decision-making
  - Role identification and assigment
  - Individual and team performance assessment
- Project management
  - Scheduling and tracking
  - Project management tools
  - Cost/benefit analysis
- Software measurement and estimation techniques
- Software quality assurance and the role of measurements
- Risk
  - The role of risk in the lifecycle
  - Risk categories including security, safety, market, financial, technology, people, quality, structure and process
- System-wide approach to risk including hazards associated with tools

Learning Outcomes

- Discuss common behaviors that contribute to the effective functioning of a team [Usage]
- Create and follow an agenda for a team meeting [Usage]
- Identify and justify necessary roles in a software development team [Usage]
- Understand the sources, hazards, and potential benefits of team conflict [Usage]
- Apply a conflict resolution strategy in a team setting [Usage]
- Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage]
- List several examples of software risks [Usage]
- Describe the impact of risk in a software development lifecycle [Usage]
- Describe different categories of risk in software systems [Usage]
- Demonstrate through involvement in a team project the central elements of team building and team management [Usage]
- Describe how the choice of process model affects team organizational structures and decision-making processes [Usage]
- Create a team by identifying appropriate roles and assigning roles to team members [Usage]
- Assess and provide feedback to teams and individuals on their performance in a team setting [Usage]
- Using a particular software process, describe the aspects of a project that need to be planned and monitored, (eg, estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management) [Usage]
- Track the progress of some stage in a project using appropriate project metrics [Usage]
- Compare simple software size and cost estimation techniques [Usage]
- Use a project management tool to assist in the assignment and tracking of tasks in a software development project [Usage]
- Describe the impact of risk tolerance on the software development process [Usage]
- Identify risks and describe approaches to managing risk (avoidance, acceptance, transference, mitigation), and characterize the strengths and short-

Unit 3: (14)	
Competences Expected: c,d,i,j,m	
Topics	**Learning Outcomes**
• Administration of the service as a practice. • Service life cycle. • Definitions and generic concepts. • Models and key principles. • Processes. • Technology and architecture. • Competence and training.	• Use and apply ITIL correctly in the software process. [Usage]
Readings : [Som17], [PM15]	

Unit 4: (14)	
Competences Expected: c,d,i,j,m	
Topics	**Learning Outcomes**
• Fundamentals and Introduction. • Control and IT Governance Frameworks.	• Use and apply COBIT correctly in the software process. [Usage]
Readings : [Som17], [PM15]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T3. Bioinformatic Algorithms (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T4. Computational Genetics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CB309. Bioinformatics (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS212. Analysis and Design of Algorithms. (5^{th} Sem) • MA307. Mathematics applied to computing . (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• Review of organic chemistry: molecules and macromolecules, sugars, nucleic acids, nucleotides, RNA, DNA, proteins, amino acids and levels of structure in proteins. • The Dogma of Life: From DNA to Proteins, Transcription, Translation, Protein Synthesis. • Genome study: Maps and sequences, specific techniques	• Achive a general knowledge of the most important topics in Molecular Biology. [Familiarity] • Understand that biological problems are a challenge to the computational world. [Assessment]
Readings : [CB00], [SM97]	

Unit 2: Sequence Comparison (4)	
Competences Expected: CS2	
Topics	**Learning Outcomes**
• Sequences of nucleotides and amino acid sequences. • Sequence alignment, paired alignment problem, exhaustive search, Dynamic programming, global alignment, local alignment, gaps penalty • Comparison of multiple sequences: sum of pairs, complexity analysis by dynamic programming, alignment heuristics, star algorithm, progressive alignment algorithms.	• Understand and solve the problem of aligning a pair of sequences. [Usage] • Understand and solve the problem of multiple sequence alignment. [Usage] • Know the various algorithms for aligning existing sequences in the literature . [Familiarity]
Readings : [CB00], [SM97], [Pev00]	

Unit 3: Phylogenetic Trees (4)	
Competences Expected: CS2	
Topics	**Learning Outcomes**
• Phylogeny: Introduction and phylogenetic relations • Phylogenetic trees: definition, type of trees, problem of search and reconstruction of trees • Reconstruction methods: parsimony methods, distance methods, maximum likelihood methods, confidence of reconstructed trees	• Understand the concept of phylogeny, phylogenetic trees and the methodological difference between biology and molecular biology. [Familiarity] • Understand the problem of the reconstruction of phylogenetic trees, to know and apply the main algorithms for the reconstruction of phylogenetic trees. [Assessment]
Readings : [CB00], [SM97], [Pev00]	

<table>
<tr><td colspan="2">Unit 4: DNA Sequence Assembling (4)</td></tr>
<tr><td colspan="2">Competences Expected: CS2</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>Biological basis: ideal case, difficulties, alternative methods for DNA sequencing
Formal Assembly Models: Shortest Common Superstring, Reconstruction, Multicontig
Algorithms for sequence assembly: representation of overlaps, paths to create superstrings, voracious algorithm, acyclic graphs.
Assembly heuristics: search for overlays, ordering fragments, alignments and consensus.</td><td>Understand the computational challenge of the Sequence Assembly problem. [Familiarity]
Understand the principle of formal model for assembly. [Assessment]
Know the main heuristics for the problem of assembjale of DNA sequences[Usage]</td></tr>
<tr><td colspan="2">Readings : [SM97], [Alu06]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 5: Secondary and tertiary structures (4)</td></tr>
<tr><td colspan="2">Competences Expected: CS2</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>Molecular structures: primary, secondary, tertiary, quaternary.
Prediction of secondary structures of RNA: formal model, pair energy, structures with independent bases, solution with Dynamic Programming, structures with loops.
Protein folding: Estructuras en proteinas, problema de protein folding.
Protein Threading: Definitions, Branch Bound Algorithm, Branch Bound for protein threading.
Structural Alignment: Definitions, DALI algorithm</td><td>Know the protein structures and the necessity of computational methods for the prediction of the geometry. [Familiarity]
Know the algorithms for solving prediction problems of secondary structures RNA, and structures in proteins. [Assessment]</td></tr>
<tr><td colspan="2">Readings : [SM97], [CB00], [Alu06]</td></tr>
</table>

Unit 6: Probabilistic Models in Molecular Biology (4)	
Competences Expected: CS2	
Topics	**Learning Outcomes**
• Probability: Random Variables, Markov Chains, Metropoli-Hasting Algorithm, Markov Random Fields, and Gibbs Sampler, Maximum Likelihood. • Hidden Markov Models (HMM), parameter estimation, Viterbi algorithm and Baul-Welch method, Application in paired and multiple alignments, Motifs detection in proteins, in eukaryotic DNA, in sequences families. • Probabilistic phylogeny: probabilistic models of evolution, likelihood of alignments, likelihood for inference, comparison of probailistic and non-probabilistic methods	• Review concepts of Probabilistic Models and understand their importance in Computational Molecular Biology. [Assessment] • Know and apply Hidden Markov Models for various analyzes in Molecular Biology.. [Usage] • Know the application of probabilistic models in Phylogeny and to compare them with non-probabilistic models[Assessment]
Readings : [Dur+98], [CB00], [Alu06], [Kro+94]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Alu06] Srinivas Aluru, ed. *Handbook of Computational Molecular Biology*. Computer and Information Science Series. Boca Raton, FL: Chapman & Hall, CRC, 2006.

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.

[Dur+98] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998, p. 357. ISBN: 9780521629713.

[Kro+94] Anders Krogh et al. “Hidden Markov Models in Computational Biology, Applications to Protein Modeling”. In: *J Molecular Biology* 235 (1994), pp. 1501–1531.

[Pev00] Pavel A. Pevzner. *Computational Molecular Biology: an Algorithmic Approach*. Cambridge, Massachusetts: The MIT Press, 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

ET301. Entrrepreneurship II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ET201. Entrrepreneurship I. (8^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The aim of this course is to provide the future professional with knowledge, attitudes and skills that will enable him/her to form his/her own software development and/or IT consultancy company. The course is divided into three units: Project Assessment, Services Marketing and Negotiations. In the first unit, the student will be able to analyze and make decisions regarding the viability of a project and/or business.

In the second unit, the aim is to prepare the student to carry out a satisfactory marketing plan of the good or service that his company can offer to the market. The third unit seeks to develop the negotiating skills of the participants through experiential and practical training and theoretical knowledge that will allow them to close contracts where both the client and the supplier are winners. We consider these issues to be extremely critical in the launch, consolidation and eventual re-launching stages of a technology-based company.

5. GOALS

- That the student understands and applies the terminology and fundamental concepts of economic engineering that allow him/her to value a project in order to make the best economic decision.

- That the student acquires the bases to form his own technology-based company.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (20)	
Competences Expected: C19	
Topics	**Learning Outcomes**
• Introduction. • Decision-making process. • The value of money over time. • Interest Rate and Rate of Return. • Simple interest and compound interest. • Cost identification. • Net Cash Flow. • Return on Investment (ROI). • Net Present Value (NPV). • Project Valuation.	• To allow the student to make decisions on how best to invest the available funds, based on the analysis of both economic and non-economic factors that determine the viability of a venture. [Assessment]
Readings : [BT06]	

Unit 2: (30)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Introduction. • Importance of marketing in service companies. • The Strategic Process. • The Marketing Plan. • Strategic marketing and operational marketing. • Segmentation, targeting and positioning of services in competitive markets. • Product life cycle. • Aspects to be considered in the setting of prices in services. • The role of advertising, sales and other forms of communication. • Consumer behaviour in services. • Fundamentals of Service Marketing. • Creation of the service model. • Service quality management.	• Brindar las herramientas al alumno para que pueda identificar, analizar y aprovechar las oportunidades de marketing que generan valor en un emprendimiento. [Usage] • To achieve that the student knows, understands and identifies criteria, abilities, methods and procedures that allow an adequate formulation of marketing strategies in specific sectors and media such as a technology-based company. [Usage]
Readings : [KK06], [LW09]	

Unit 3: (10)	
Competences Expected: C18	
Topics	**Learning Outcomes**
• Introduction. What is a negotiation?. • Theory of negotiation needs. • The negotiation process. • Trading styles. • Game theory. • The Harvard method of negotiation.	• Know the key points in the negotiation process. [Usage] • Establish an effective negotiation methodology. [Usage] • To develop skills and abilities that allow to carry out a successful negotiation. [Usage]
Readings : [FUP96], [MM06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[BT06] Leland Blank and Anthony Tarkin. *Ingeniería Económica*. McGraw Hill, México D.F., México, 2006.

[FUP96] Roger Fisher, William Ury, and Bruce Patton. *Si... ¡de acuerdo! Cómo negociar sin ceder*. Norma, Barcelona, 1996.

[KK06] Philip Kotler and Kevin L. Keller. *Dirección de Marketing*. Prentice Hall, México, 2006.

[LW09] Christopher Lovelock and Jochen Wirtz. *Marketing de servicios. Personal, tecnología y estratégia*. Prentice Hall, México, 2009.

[MM06] Fernando de Manuel Dasí and Rafael Martínez-Vilanova Martínez. *Técnicas de Negociación. Un método práctico*. Esic, Madrid, 2006.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS365. Evolutionary Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

La Computación Evolutiva comprende un conjunto de metodologías de búsqueda y optimización cuya base primordial es el Paradigma Neodarwiniano que agrupa la Herencia Genética (Mendel), el Seleccionismo (Weismann) y la Evolución de las Especies (Darwin) que, cuando llevadas a implementaciones computacionales, ofrecen una herramienta poderosa de optimización global para una determinada función objetivo. Son bastante robustos cuando se supone la existencia de muchos óptimos locales. De esta forma, estos algoritmos pueden aplicarse en diversos problemas de optimización.

5. GOALS

- Que el alumno sea capaz de entender y aplicar el Paradigma Neodarwiniano para solucionar problemas complejos de optimización.
- Entendimiento a detalle del principio, fundamentos teóricos, funcionamiento, implementación, intepretación de resultados y operación de los algoritmos de la Computación Evolutiva más populares y utilizados por la comunidad científica y profesional.
- Conocimiento del estado del arte en Computación Evolutiva
- Capacidad de tratar un problema real de optimización utilizando Computación Evolutiva

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introducción a la Optimización (4)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Definiciones de Optimización: principio de estabilidad, optimización global. • Optimización Clásica: Definición del problema de optimización, concepto de convexidad, optimización numérica y combinatoria. • Técnicas de optimización clásica: optimización lineal, algoritmo simplex, optimización no lineal, algoritmos *steepest descent, conjugate gradient*, algoritmos de búsqueda, programación dinámica, • Heurísticas: definición, *Tabu search*, *Hill Climbing Simulated Annealing*, *Evolutionary Algorithms*	• Entender los principios básicos de la optimización • Entender e implementar algoritmos básicos de Optimización aplicados a problemas *benchmark.* • Entender la necesidad de uso de heurísticas
Readings : [Wei09], [RBK12]	

Unit 2: Computación Evolutiva: Conceptos básicos (8)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Computación Evolutiva: definiciones • Ideas precursoras: El origen de las ideas, L'Eclerc, Lamarck, Darwin, Weismann, Mendel, Baldwin, Paradigma Neodarwiniano • Conceptos básicos de Computación Evolutiva: genes, cromosomas, individuos, población. • Paradigmas de la Computación Evolutiva: Programación Evolutiva, Estrategias Evolutivas, Algoritmos Genéticos, *Learning Classifier Systems*, Programación Genética.	• Entender los principios básicos que rigen la computación evolutiva • Conocer el contexto en que surgió la computación evolutiva.
Readings : [RBK12], [Wei09], [Fog95], [**koza98**], [Mit04], [Mic96]	

Unit 3: Algoritmo Genético Canónico (8)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Algoritmo Genético: definición, componentes. • Algoritmo Genético Canónico: procedimiento elemental, ciclo de un AG, representación (codificación binaria, real a binario, decodificación binario a real), inicialización de la población, evaluación y aptitud, selección (proporcional, torneo), operadores genéticos (cruces, mutaciones), el dilema *exploiting-exploring*, ajustes en la aptitud, ajustes en la selección. • Monitoreo de un AG: curvas *best-so-far, online, off-line* • Convergencia • Teoría de *Schemata*: Máscaras, esquemas, definiciones y propiedades, *Schemata theorem*: impacto de la selección, cruce de 1 punto y mutación, teorema fundamental de los algoritmos genéticos, hipótesis de los bloques constructores.	• Entender los algoritmos genéticos tradicionales. • Analizar y evaluar ventajas y desventajas del modelo genético tradicional. • Implementar un ejemplo de algoritmo genético tradicional y analizar su comportamiento.
Readings : [RBK12], [Hol75], [Gol89], [Mit04], [Mic96]	

Unit 4: Algoritmos Evolutivos en Optimización Numérica (8)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Problemas con restricciones: definiciones, espacios válido e inválido. • Tratamiento de las restricciones: Penalización, reparación, uso de codificadores, operadores especializados. • Uso de codificación real: binario vs. real, algoritmo evolutivo con codificación real. • Modelo GENOCOP: tratamiento de restricciones lineales, inicialización, operadores, inicialización, modelo GENOCOP III para restricciones no lineales: reparación de individuos.	• Comprensión de las formas de tratar problemas de optimización con restricciones. • Entender y analizar los algoritmos evolutivos con codificación real. • Evaluar la aplicación de computación evolutiva en problemas de optimización numérica
Readings : [RBK12], [Mic96], [Mic00], [SC00]	

Unit 5: Algoritmos Evolutivos en Optimización Combinatoria (8)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Espacios discretos y finitos • Algoritmos Evolutivos discretos: definición, modelo discreto generalizado • Algoritmos Evolutivos de orden: representación de soluciones, operadores de orden: cruces, mutaciones • Aplicaciones: *Quadratic assignment Problem* – QAP, *Travelling Salesman Problem* – TSP • Problemas de Planificación: variables típicas, carácteristicas, representación, codificadores, evaluación de una planificación.	• Comprender e identificar el uso de Computación Evolutiva en problemas de optimización combinatoria • Evaluar la aplicación de computación evolutiva en problemas reales discretos
Readings : [RBK12], [Mit04], [Cru03]	

Unit 6: Paralelización y Multiobjetivos (8)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• PEA - Algoritmos Evolutivos en Paralelo: arquitecturas de paralelización, arquitecturas *master-slave*, *coarse-grained*, *fine-grained* e híbridas • Análisis de la ejecución de una implementación *master-slave*. • Optimización de Multiples Objetivos: Definición formal, criterio de Pareto, Algoritmos Evolutivos Multi Objetivos (MOEA) sin uso de Pareto, MOEA con uso de Pareto: MOGA, NSGA, NPGA, NPGA2, PESA, SPEA, SPEA-II, Algoritmo Microgenético. • MOEA – Métricas de desempeño, investigación futura	• Comprender y analizar la capacidad de paralelización de los modelos evolutivos • Analizar la aplicabilidad de Computación Evolutiva en problemas de múltiples objetivos • Implementación de modelos paralelos y multiobjetivo
Readings : [RBK12], [Can00], [Coe07]	

Unit 7: Algoritmos Genéticos Avanzados (16)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• HEA – Algoritmos Evolutivos Híbridos: Por qué hibridizar?, formas de hibridización, búsqueda local y aprendizaje. • GP – Programación Genética: definición, representación, ciclo de la GP. • CA – Algoritmos Culturales: Evolución Cultural, componentes, procedimiento, espacio de creencia, operadores culturales. • CoEv – Coevolución: carácteristicas, modelo competitivo, modelo cooperativo. • DE – Evolución Diferencial: inicialización, operaciones, selección, DE vs. GA, variantes de DE, *Dynamic DE* • QIEA – Algoritmos Evolutivos con Inspiración Quántica: Computación quántica, algoritmos con inspiración quántica, QIEA-**B**, QIEA-**R**	• Reconocer y analizar la necesidad de usar Algoritmos Evolutivos más avanzados • Implementación de modelos avanzados de computación evolutiva
Readings : [RBK12], [ElM+06], [Koz92], [**Reynolds94**], [SP95], [Cru07]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Can00] Erick Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN: 0792372212.

[Coe07] Carlos A. Coello Coello. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. 2nd Edition. Springer, Sept. 2007.

[Cru03] André Vargas Abs da Cruz. "Otimização de planejamento com restrições de precedência usando algoritmos genéticos e co-evolução cooperativa". MA thesis. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Feb. 2003. URL: `http://www2.dbd.puc-rio.br/pergamum/biblioteca/php/mostrateses.php?open=1&arqtese=5000066121_03_Indice.html`.

[Cru07] André Abs da Cruz. "Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica". (In Portuguese). PhD thesis. Rio de Janeiro, Brasil: Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Mar. 2007.

[ElM+06] Tarek A. El-Mihoub et al. "Hybrid Genetic Algorithms: A Review". In: *Engineering Letters* 13.2 (Aug. 2006). ISSN: 1816-0948. URL: `www.engineeringletters.com/issues_v13/issue.../EL_13_2_11.pdf`.

[Fog95] David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence.* New York: The Institute of Electrical and Electronic Engineers, 1995.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, Massachusetts: Addison-Wesley Publishing Co., 1989.

[Hol75] John Henry Holland. *Adaptation in Natural and Artificial Systems.* first. Ann Arbor, Michigan: University of Michigan Press, 1975.

[Koz92] John R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* Cambridge, Massachusetts: The MIT Press, 1992.

[Mic00] Zbigniew Michalewicz. "Introduction to constraint-handling tecniques, Decoders, Repair algorithms, Constraint-preserving operators". In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 38–40, 49–55, 56–61, 62–68.

[Mic96] Zibgniew Michalewicz. *Genetic Algorithms+Data Structures = Evolution Programs.* Springer-Verlag, 1996.

[Mit04] Melanie Mitchell. *An Introduction to Genetic Algorithms: Complex Adaptative Systems.* The MIT Press, 2004.

[RBK12] Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, eds. *Handbook of Natural Computing.* 1st. Springer Publishing Company, Incorporated, 2012. ISBN: 3540929096, 9783540929093.

[SC00] Alice E. Smith and David W. Coit. "Penalty functions". In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 41–48.

[SP95] Rainer Storn and Kenneth Price. *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces.* Tech. rep. TR-95-012. Berkeley, California: International Computer Science Institute, Mar. 1995.

[Wei09] Thomas Weise. *Global Optimization Algorithms - Theory and Application.* http://www.it-weise.de. 2009.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3P2. Cloud Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS370. Big Data. (9^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In order to understand the advanced computational techniques, the students must have a strong knowledge of the various discrete structures, structures that will be implemented and used in the laboratory in the programming language.

5. GOALS

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student apply efficient travel strategies to be able to search data in an optimal way.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Distributed Systems (15)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Faults (cross-reference OS/Fault Tolerance)
– Network-based (including partitions) and node-based failures
– Impact on system-wide guarantees (e.g., availability)
• Distributed message sending
– Data conversion and transmission
– Sockets
– Message sequencing
– Buffering, retrying, and dropping messages
• Distributed system design tradeoffs
– Latency versus throughput
– Consistency, availability, partition tolerance
• Distributed service design
– Stateful versus stateless protocols and services
– Session (connection-based) designs
– Reactive (IO-triggered) and multithreaded designs
• Core distributed algorithms
– Election, discovery
</td><td>
• Distinguish network faults from other kinds of failures [Familiarity]
• Explain why synchronization constructs such as simple locks are not useful in the presence of distributed faults [Familiarity]
• Write a program that performs any required marshalling and conversion into message units, such as packets, to communicate interesting data between two hosts [Usage]
• Measure the observed throughput and response latency across hosts in a given network [Usage]
• Explain why no distributed system can be simultaneously consistent, available, and partition tolerant [Familiarity]
• Implement a simple server – for example, a spell checking service [Usage]
• Explain the tradeoffs among overhead, scalability, and fault tolerance when choosing a stateful v stateless design for a given service [Familiarity]
• Describe the scalability challenges associated with a service growing to accommodate many clients, as well as those associated with a service only transiently having many clients [Familiarity]
• Give examples of problems for which consensus algorithms such as leader election are required [Usage]
</td></tr>
<tr><td colspan="2">Readings : [Cou+11]</td></tr>
</table>

Unit 2: Cloud Computing (15)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Visión global de *Cloud Computing.*
• Historia.
• Visión global de las tecnologias que envuelve.
• Beneficios, riesgos y aspectos económicos.
• Cloud services
  – Infrastructure as a service
    ∗ Elasticity of resources
    ∗ Platform APIs
  – Software as a service
  – Security
  – Cost management
• Internet-Scale computing
  – Task partitioning
  – Data access
  – Clusters, grids, and meshes
	• Explicar el concepto de Cloud Computing. [Familiarity]
	• Listar algunas tecnologias relacionadas con Cloud Computing. [Familiarity]
	• Explain strategies to synchronize a common view of shared data across a collection of devices [Familiarity]
	• Discutir las ventajas y desventajas del paradigma de Cloud Computing. [Familiarity]
	• Expresar los beneficios económicos así como las carácteristicas y riesgos del paradigma de Cloud para negocios y proveedores de cloud. [Familiarity]
	• Diferenciar entre los modelos de servicio. [Usage]

Readings : [HDF11], [BVS13]	

Unit 3: Centros de Procesamiento de Datos (10)	
Competences Expected: g,i	
Topics	**Learning Outcomes**
• Visión global de un centro de procesamiento de datos. • Consideraciones en el diseño. • Comparación de actuales grandes centros de procesamiento de datos.	• Describir la evolución de los Data Centers. [Familiarity] • Esbozar la arquitectura de un data center en detalle. [Familiarity] • Indicar consideraciones de diseño y discutir su impacto. [Familiarity]
Readings : [HDF11], [BVS13]	

<table>
<tr><td colspan="2">Unit 4: Cloud Computing (20)</td></tr>
<tr><td colspan="2">Competences Expected: i,j</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Virtualization
  – Shared resource management
  – Migration of processes
• Seguridad, recursos y isolamiento de fallas.
• Almacenamiento como servicio.
• Elasticidad.
• Xen y WMware.
• Amazon EC2.
</td><td>
• Virtualization
  – Shared resource management
  – Migration of processes
. [Familiarity]
• Explain the advantages and disadvantages of using virtualized infrastructure. [Familiarity]
• Identificar las razones por qué la virtualización está llegando a ser enormente útil, especialmente en la cloud. [Familiarity]
• Explicar diferentes tipos de isolamiento como falla, recursos y seguridad proporcionados por la virtualización y utilizado por la cloud. [Familiarity]
• Explicar la complejidad que puede tener el administrar en términos de niveles de abstracción y interfaces bien definidas y su aplicabilidad para la virtualización en la cloud. [Familiarity]
• Definir virtualización y identificar diferentes tipos de máquinas virtuales. [Familiarity]
• Identificar condiciones de virtualización de CPU, reconocer la diferencia entre full virtualization y paravirtualization, explicar emulación como mayor técnica para virtualización del CPU y examinar planificación virtual del CPU en Xen. [Familiarity]
• Esbozar la diferencia entre la clásica memoria virtual del SO y la virtualización de memoria. Explicar los múltiplos niveles de mapeamiento de páginas en oposición a la virtualización de la memoria. Definir memoria over-commitment e ilustrar sobre WMware memory ballooning como técnica de reclamo para sistemas virtualizados con memoria over-committed. [Familiarity]
</td></tr>
<tr><td colspan="2">Readings : [HDF11], [BVS13]</td></tr>
</table>

Unit 5: Cloud Computing (12)	
Competences Expected: i,j	
Topics	**Learning Outcomes**

• Cloud-based data storage
  – Shared access to weakly consistent data stores
  – Data synchronization
  – Data partitioning
  – Distributed file systems
  – Replication
• Visión global sobre tecnologías de almacenamiento.
• Conceptos fundamentales sobre almacenamiento en la cloud.
• Amazon S3 y EBS.
• Sistema de archivos distribuidos.
• Sistema de bases de datos NoSQL.
	• Describir la organización general de datos y almacenamiento. [Familiarity]
	• Identificar los problemas de escalabilidad y administración de la big data. Discutir varias abstracciones en almacenamiento. [Familiarity]
	• Comparar y contrastar diferentes tipos de sistema de archivos. Comparar y contrastar el Sistema de Archivos Distribuido de Hadoop (HDFS) y el Sistema de Archivos Paralelo Virtual (PVFS). [Usage]
	• Comparar y contrastar diferentes tipos de bases de datos. Discutir las ventajas y desventajas sobre las bases de datos NoSQL. [Usage]
	• Discutir los conceptos de almacenamiento en la cloud. [Familiarity]

Readings : [HDF11], [BVS13]	

Unit 6: Modelos de Programación (12)	
Competences Expected: g,j	
Topics	**Learning Outcomes**
• Visión global de los modelso de programación basados en cloud computing. • Modelo de Programación MapReduce. • Modelo de programación para aplicaciones basadas en Grafos.	• Explicar los aspectos fundamentales de los modelos de programación paralela y distribuida. [Familiarity] • Diferencias entre los modelos de programación: MapReduce, Pregel, GraphLab y Giraph. [Usage] • Explicar los principales conceptos en el modelo de programación MapReduce. [Usage]
Readings : [HDF11], [BVS13], [Low+12], [Mal+10], [Bal+08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: http://doi.acm.org/10.1145/1367497.1367618.

[BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.

[Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.

[HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.

[Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: http://dx.doi.org/10.14778/2212351.2212354.

[Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *Proc. ACM SIGMOD*. SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: http://doi.acm.org/10.1145/1807167.1807184.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3P3. Internet of Things (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS3P1. Parallel and Distributed Computing . (8^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The last decade has an explosive growth in multiprocessor computing, including multi-core processors and distributed data centers. As a result, parallel and distributed computing has evolved from a broadly elective subject to be one of the major components in mesh studies in undergraduate computer science. Both parallel computing and distribution involve the simultaneous execution of multiple processes on different devices that change position.

5. GOALS

- That the student is able to create parallel applications of medium complexity by efficiently taking advantage of different mobile devices.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Parallelism Fundamentals (18)	
Competences Expected: a	
Topics	**Learning Outcomes**

• Multiple simultaneous computations
• Goals of parallelism (e.g., throughput) versus concurrency (e.g., controlling access to shared resources)
• Parallelism, communication, and coordination
  – Parallelism, communication, and coordination
  – Need for synchronization
• Programming errors not found in sequential programming
  – Data races (simultaneous read/write or write/write of shared state)
  – Higher-level races (interleavings violating program intention, undesired non-determinism)
  – Lack of liveness/progress (deadlock, starvation)
	• Distinguish using computational resources for a faster answer from managing efficient access to a shared resource [Familiarity]
	• Distinguish multiple sufficient programming constructs for synchronization that may be inter-implementable but have complementary advantages [Familiarity]
	• Distinguish data races from higher level races [Familiarity]

Readings : [Pac11], [Mat14], [Qui03]	

Unit 2: Parallel Architecture (12)	
Competences Expected: b	
Topics	**Learning Outcomes**

• Multicore processors
• Shared vs distributed memory
• Symmetric multiprocessing (SMP)
• SIMD, vector processing
• GPU, co-processing
• Flynn's taxonomy
• Instruction level support for parallel programming
  – Atomic instructions such as Compare and Set
• Memory issues
  – Multiprocessor caches and cache coherence
  – Non-uniform memory access (NUMA)
• Topologies
  – Interconnects
  – Clusters
  – Resource sharing (e.g., buses and interconnects)
	• Explain the differences between shared and distributed memory [Assessment]
	• Describe the SMP architecture and note its key features [Assessment]
	• Characterize the kinds of tasks that are a natural match for SIMD machines [Usage]
	• Describe the advantages and limitations of GPUs vs CPUs [Usage]
	• Explain the features of each classification in Flynn's taxonomy [Usage]
	• Describe the challenges in maintaining cache coherence [Familiarity]
	• Describe the key performance challenges in different memory and distributed system topologies [Familiarity]

Readings : [Pac11], [KH13], [SK10]	

Unit 3: Parallel Decomposition (18)	
Competences Expected: i	
Topics	**Learning Outcomes**

• Need for communication and coordination/synchronization
• Independence and partitioning
• Basic knowledge of parallel decomposition concept
• Task-based decomposition
  – Implementation strategies such as threads
• Data-parallel decomposition
  – Strategies such as SIMD and MapReduce
• Actors and reactive processes (e.g., request handlers)
	• Explain why synchronization is necessary in a specific parallel program [Usage]
	• Identify opportunities to partition a serial program into independent parallel modules [Familiarity]
	• Write a correct and scalable parallel algorithm [Usage]
	• Parallelize an algorithm by applying task-based decomposition [Usage]
	• Parallelize an algorithm by applying data-parallel decomposition [Usage]
	• Write a program using actors and/or reactive processes [Usage]

Readings : [Pac11], [Mat14], [Qui03]	

Unit 4: Communication and Coordination (18)	
Competences Expected: i	
Topics	**Learning Outcomes**

• Shared Memory
• Consistency, and its role in programming language guarantees for data-race-free programs
• Message passing
  – Point-to-point versus multicast (or event-based) messages
  – Blocking versus non-blocking styles for sending and receiving messages
  – Message buffering (cross-reference PF/Fundamental Data Structures/Queues)
• Atomicity
  – Specifying and testing atomicity and safety requirements
  – Granularity of atomic accesses and updates, and the use of constructs such as critical sections or transactions to describe them
  – Mutual Exclusion using locks, semaphores, monitors, or related constructs
    ∗ Potential for liveness failures and deadlock (causes, conditions, prevention)
  – Composition
    ∗ Composing larger granularity atomic actions using synchronization
    ∗ Transactions, including optimistic and conservative approaches
• Consensus
  – (Cyclic) barriers, counters, or related constructs
• Conditional actions
  – Conditional waiting (e.g., using condition variables)
	• Use mutual exclusion to avoid a given race condition [Usage]
	• Give an example of an ordering of accesses among concurrent activities (eg, program with a data race) that is not sequentially consistent [Familiarity]
	• Give an example of a scenario in which blocking message sends can deadlock [Usage]
	• Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity]
	• Write a program that correctly terminates when all of a set of concurrent tasks have completed [Usage]
	• Give an example of a scenario in which an attempted optimistic update may never complete [Familiarity]
	• Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]

Readings : [Pac11], [Mat14], [Qui03]	

<table>
<tr><td colspan="2">Unit 5: Parallel Algorithms, Analysis, and Programming (18)</td></tr>
<tr><td colspan="2">Competences Expected: i</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>
• Critical paths, work and span, and the relation to Amdahl's law
• Speed-up and scalability
• Naturally (embarrassingly) parallel algorithms
• Parallel algorithmic patterns (divide-and-conquer, map and reduce, master-workers, others)
  – Specific algorithms (e.g., parallel MergeSort)
• Parallel graph algorithms (e.g., parallel shortest path, parallel spanning tree) (cross-reference AL/Algorithmic Strategies/Divide-and-conquer)
• Parallel matrix computations
• Producer-consumer and pipelined algorithms
• Examples of non-scalable parallel algorithms
</td><td>
• Define “critical path”, “work”, and “span” [Familiarity]
• Compute the work and span, and determine the critical path with respect to a parallel execution diagram [Usage]
• Define “speed-up” and explain the notion of an algorithm's scalability in this regard [Familiarity]
• Identify independent tasks in a program that may be parallelized [Usage]
• Characterize features of a workload that allow or prevent it from being naturally parallelized [Familiarity]
• Implement a parallel divide-and-conquer (and/or graph algorithm) and empirically measure its performance relative to its sequential analog [Usage]
• Decompose a problem (eg, counting the number of occurrences of some word in a document) via map and reduce operations [Usage]
• Provide an example of a problem that fits the producer-consumer paradigm [Usage]
• Give examples of problems where pipelining would be an effective means of parallelization [Usage]
• Implement a parallel matrix algorithm [Usage]
• Identify issues that arise in producer-consumer algorithms and mechanisms that may be used for addressing them [Usage]
</td></tr>
<tr><td colspan="2">Readings : [Mat14], [Qui03]</td></tr>
</table>

Unit 6: Parallel Performance (18)	
Competences Expected: j	
Topics	**Learning Outcomes**

• Load balancing
• Performance measurement
• Scheduling and contention (cross-reference OS/Scheduling and Dispatch)
• Evaluating communication overhead
• Data management
  – Non-uniform communication costs due to proximity (cross-reference SF/Proximity)
  – Cache effects (e.g., false sharing)
  – Maintaining spatial locality
• Power usage and management
	• Detect and correct a load imbalance [Usage]
	• Calculate the implications of Amdahl's law for a particular parallel algorithm (cross-reference SF/Evaluation for Amdahl's Law) [Usage]
	• Describe how data distribution/layout can affect an algorithm's communication costs [Familiarity]
	• Detect and correct an instance of false sharing [Usage]
	• Explain the impact of scheduling on parallel performance [Familiarity]
	• Explain performance impacts of data locality [Familiarity]
	• Explain the impact and trade-off related to power usage on parallel performance [Familiarity]

Readings : [Pac11], [Mat14], [KH13], [SK10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

[Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: `http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf`.

[Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.

[Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS404. Final Project III (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	6
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS403. Final Project II. (9^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to enable students to complete properly their draft of thesis.

5. GOALS

- That the student completes this course with his thesis elaborated in sufficient quality as for an immediate support.
- That the student formally present the draft dissertation before the authorities of the faculty
- The deliverables of this course are:

  Parcial: Advancement of the thesis project including in the document: introduction, theoretical framework, state of the art, proposal, analysis and / or experiments and solid bibliography.

  Final: Full thesis document and ready to support in a period of no more than fifteen days.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Escritura del Borrador del trabajo de final de carrera (tesis) (60)	
Competences Expected: h,g,e,f,i,l	
Topics	**Learning Outcomes**
• Writing and correction of the work of end of career	• Experimental part completed (if appropriate to the project) [Assessment] • Verify that the document complies with the thesis format of the course [Assessment] • Delivery of the completed thesis draft and considered ready for public support (approval requirement)[Assessment]
Readings : [IEE08], [Ass08], [Cit08]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Ass08] Association for Computing Machinery. *Digital Libray.* http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray.* http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray.* http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS366. Robotics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS261. Intelligent Systems. (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

That the student knows and understands the concepts and fundamental principles of control, road planning and the definition of strategies in robotics as well as concepts of robotic perception in a way that understands the potential of robotic systems

5. GOALS

- Synthesize the potential and limitations of the state-of-the-art of today's robotic systems.
- Implement Simple Motion Planning Algorithms.
- Explain the uncertainties associated with sensors and how to treat them.
- Designing a Simple Control Architecture.
- Describes several navigation strategies
- Describe the importance of recognizing images and objects in intelligent systems
- Outline the main techniques of object recognition
- Describe the different characteristics of the technologies used in perception

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Robotics (5)	
Competences Expected: a,b	
Topics	**Learning Outcomes**

• Overview: problems and progress
  – State-of-the-art robot systems, including their sensors and an overview of their sensor processing
  – Robot control architectures, e.g., deliberative vs. reactive control and Braitenberg vehicles
  – World modeling and world models
  – Inherent uncertainty in sensing and in control
• Configuration space and environmental maps
	• List capabilities and limitations of today's state-of-the-art robot systems, including their sensors and the crucial sensor processing that informs those systems [Familiarity]
	• Integrate sensors, actuators, and software into a robot designed to undertake some task [Usage]

Readings : [**Siegwart04**], [**Trun05**], [**Stone00**]	

Unit 2: Robotics (15)	
Competences Expected: a,b,i,h	
Topics	**Learning Outcomes**
• Interpreting uncertain sensor data • Localizing and mapping	• Program a robot to accomplish simple tasks using deliberative, reactive, and/or hybrid control architectures [Usage] • Implement fundamental motion planning algorithms within a robot configuration space [Usage]
Readings : [**Siegwart04**], [**Trun05**]	

Unit 3: Robotics (20)	
Competences Expected: h,i	
Topics	**Learning Outcomes**
• Navigation and control • Motion planning	• Characterize the uncertainties associated with common robot sensors and actuators; articulate strategies for mitigating these uncertainties [Usage] • List the differences among robots' representations of their external environment, including their strengths and shortcomings [Usage]
Readings : [**Siegwart04**]	

Unit 4: Perception and Computer Vision (10)	
Competences Expected: a,b,c,f	
Topics	**Learning Outcomes**
• Computer vision – Image acquisition, representation, processing and properties – Shape representation, object recognition and segmentation – Motion analysis • Modularity in recognition	• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Usage] • Implement 2d object recognition based on contour- and/or region-based shape representations [Usage]
Readings : [**Sonka07**], [**Gonzales07**]	

Unit 5: Robotics (10)	
Competences Expected: a,b,i,h	
Topics	**Learning Outcomes**
• Multiple-robot coordination	• Compare and contrast at least three strategies for robot navigation within known and/or unknown environments, including their strengths and shortcomings [Familiarity] • Describe at least one approach for coordinating the actions and sensing of several robots to accomplish a single task [Familiarity]
Readings : [**Stone00**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS369. Topics in Artificial Intelligence (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS262. Machine learning. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It provides a set of tools to solve problems that are difficult to solve with traditional algorithmic methods. Including heuristics, planning, formalisms in the representation of knowledge and reasoning, machine learning techniques, techniques applicable to action and reaction problems: as well as the learning of natural language, artificial vision and robotics among others.

5. GOALS

- Take an advanced course in Artificial Intelligence suggested by the ACM/IEEE curriculum.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (60)	
Competences Expected: a,h	
Topics	**Learning Outcomes**
• Intelligent Systems. • Automated Reasoning. • Knowledge Based Systems. • Machine Learning. [RN03],[Hay99] • Planning Systems. • Natural Language Processing. • Agents. • Robotics. • Symbolic Computing. • Genetic Algorithms. [Gol89]	• To deepen in several techniques related to Artificial Intelligence. [Usage]
Readings : [RN03], [Hay99], [Gol89]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[Hay99] Simon Haykin. *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[RN03] Stuart Russell and Peter Norvig. *Inteligencia Artifical: Un enfoque moderno*. Prentice Hall, 2003.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS374. Text Processing for Data Science (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2702. Data Management II. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Usage] • ... [Usage] • ... [Usage]
Readings : [Bur04], [Cel05]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS379. Tópicos Avanzados en Ciencia de Datos (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2702. Data Management II. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

5. GOALS

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Physical Database Design (10)	
Competences Expected: b,j	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Usage] • ... [Usage] • ... [Usage]
Readings : [Bur04], [Cel05]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T5. Modeling and Simulation of Biological Systems (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS3T9. Advanced Topics in Bioinformatics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2T1. Computational Biology. (7^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

5. GOALS

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Topics	**Learning Outcomes**
• ... • ... • ...	• ... [Familiarity] • ... [Assessment]
Readings : [CB00], [SM97]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

FG211. Professional Ethics (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Ethics is a constitutive part inherent to the human being, and as such it must be reflected in the daily and professional actions of the human person.

It is indispensable that the person assumes an active role in society because the economic-industrial, political and social systems are not always in function of values and principles, being these in reality the pillars on which all the action of professionals should be based.

5. GOALS

- That the student broadens his own personal criteria for moral discernment in professional work, so that he not only takes into account the relevant technical criteria but also incorporates moral questions and adheres to correct professional ethics, so that he is capable of making a positive contribution to the economic and social development of the city, region, country and global community.[Usage]

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: (12)	
Competences Expected: C10,C21	
Topics	**Learning Outcomes**
• Be professional and be moral. • Moral objectivity and the formulation of moral principles. • The professional and his values. • The moral conscience of the person. • The contribution of the DSI to the professional's work. • The common good and the principle of subsidiarity. • Moral principles and private property. • Justice: some basic concepts.	• To present the student with the importance of having principles and values in today's society.[Usage] • To present some of the principles that could contribute to society if applied and lived day by day. [Usage] • To present to the students the contribution of the Social Doctrine of the Church in their professional work. [Usage]
Readings : [Com92], [Sch95], [Loz00], [Arg06]	

Unit 2: (12)	
Competences Expected: C20,C22	
Topics	**Learning Outcomes**
• The individual responsibility of the worker in the company. • Leadership and professional ethics in the work environment. • General principles on collaboration in immoral acts. • The professional in the face of bribery: 'victim or collaborator'.	• To present the student with the role of individual social responsibility and leadership in the company. [Familiarity] • To know the judgment of ethics in the face of corruption and bribery as a form of work relationship. [Familiarity] • To present the profession as a form of personal fulfillment, and as a consequence. []
Readings : [Com92], [Man07], [Sch95], [Pér98], [Nie03]	

Unit 3: (12)	
Competences Expected: C10,C20,C21	
Topics	**Learning Outcomes**
• Professional ethics versus general ethics. • Work and profession in the current times. • Ethics, science and technology. • Ethical values in organizations related to the use of information. • Ethical values in the Information Society era.	• To present the student with the interrelations between ethics and the disciplines of the latest technological era. [Familiarity]
Readings : [Com92], [IEE04], [Her06]	

Unit 4: (12)	
Competences Expected: C21,C22	
Topics	**Learning Outcomes**

• Computer ethics.
  – Ethics and software.
  – Free software.
• Telecommunications regulation and ethics.
  – Internet ethics.
• Copyright and patents.
• Ethics in consulting services.
• Ethics in technological innovation processes.
• Ethics in technology management and technology-based companies.
	• To present the student with some aspects that confront ethics with the work of emerging disciplines in the information society. [Familiarity]

Readings : [Com02], [Her06], [Com92]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Arg06] Argandoña. “La identidad Cristiana del Directivo de Empresa”. In: *IESE* (2006).

[Com02] Pontificio Consejo para las Comunicaciones Sociales. *Ética en Internet.* 2002.

[Com92] Association for Computing Machinery (ACM). “ACM Code of Ethics and Professional Conduct”. In: (1992). URL: `http://www.acm.org/about/code-of-ethics`.

[Her06] A. Hernández. *Ética Actual y Profesional. Lecturas para la Convivencia Global en el Siglo XXI.* Ed. Thomson, 2006.

[IEE04] IEEE. “IEEE Code of Ethics”. In: *IEE* (2004). URL: `http://www.ieee.org/about/corporate/governance/p7-8.html`.

[Loz00] C Loza. “El aporte de la Doctrina Social de la Iglesia a la Toma de Decisiones Empresariales”. In: *Separata ofrecida por el profesor* (2000).

[Man07] G. Manzone. *La Responsabilidad de la Empresa, Business Ethics y Doctrina Social de la Iglesia en Diálogo*. Universidad Católica San Pablo, 2007.

[Nie03] R. Nieburh. *El Yo Responsable. Ensayo de Filosofía Moral Cristiana*. Bilbao, 2003.

[Pér98] J. A. Pérez López. *Liderazgo y Ética en la Dirección de Empresas*. Bilbao, 1998.

[Sch95] E. Schmidt. *Ética y Negocios para América Latina*. Universidad del Pacífico, 1995.

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

ET302. Entrerpreneurship III (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ET301. Entrerpreneurship II. (9^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course is part of the training area of technology-based companies, aims to address all processes and good practices in the project management recommended by the *Project Management Institute* (PMI) contained in the *Project Management Body of Knowledge 2012* (PMBOK) applied in particular to technology-based projects such as construction, development, integration and implementation of application software.

The future professional who intends to venture into a software in the competitive globalised market, it must necessarily know the hard skills and practice the soft skills that are considered in the PMBOK. All contracts for the supply of goods (Hardware) or intangible (Software) as well as the services of consulting should be handled as small projects.

We believe it is of utmost importance to impart the fundamentals and experiences associated with project management to future professionals, we must consider that currently the client companies (national or international) that demand solutions require consulting companies to carry out system projects and information technology with PMI standards, more and more turns out to be a condition of exigibility to be able to win tenders and sign contracts for the supply of technology solutions, It also requires that the project leader, in addition to his or her training and experience to bring the project to a successful conclusion is a PMP.

5. GOALS

- That the student masters the concepts related to the management of computer projects.
- To provide the student with the techniques and tools that allow him/her to successfully manage projects of various magnitudes.
- That the student builds his business plan oriented to get an international investor who can promote and project the company to an international environment.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Conceptual Framework of Project Management (15)	
Competences Expected: C19	
Topics	**Learning Outcomes**
• Introduction. • Purpose of the PMBOK guide, 'What is a project', 'What is project management', The structure of the PMBOK guide, Areas of expertise, context of project management. • Project Life Cycle and Organization. • Project life cycle, project stakeholders, organizational influences.	• To know the conceptual framework in which the projects are developed. [Usage]
Readings : [Pro12], [Rit09]	

Unit 2: Standard for the management of a project (15)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Project Management Processes for a Project. • Project management processes, project management process groups, process interactions, correspondence of project management processes.	• Know the standards of project management applied to projects. [Usage]
Readings : [Pro12], [Rit09]	

Unit 3: Project management knowledge areas (60)	
Competences Expected: C23	
Topics	**Learning Outcomes**
• Introduction. • Project Integration Management. • Project Scope Management. • Project Time Management. • Project Cost Management. • Project Quality Management. • Project Human Resources Management. • Project Communications Management. • Project Risk Management. • Project Procurement Management.	• Understand the nature of project management and its importance to project success. [Assessment] • Acquire the necessary knowledge to successfully manage projects in terms of: Time, Costs, Scope, Risks, Quality, HR, Procurement, Communications and Integration. [Usage] • Appreciate the importance of good project management. [Assessment] • Demonstrate skills in making effective presentations. [Usage] • Develop skills to manage multidisciplinary work teams. [Usage]
Readings : [Pro12], [Rit09]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Pro12] PMI Project Management Institute. *PMBOK Guide, 5th Edition.* Project Management Institute, 2012.

[Rit09] PMP Rita Mulcahy. *PMP Exam Prep - 6th Edition.* RMC Publications, 2009.